

1999

Modeling and simulating chemical weapon dispersal patterns in DIRSIG

Peter Arnold

Follow this and additional works at: <http://scholarworks.rit.edu/theses>

Recommended Citation

Arnold, Peter, "Modeling and simulating chemical weapon dispersal patterns in DIRSIG" (1999). Thesis. Rochester Institute of Technology. Accessed from

This Thesis is brought to you for free and open access by the Thesis/Dissertation Collections at RIT Scholar Works. It has been accepted for inclusion in Theses by an authorized administrator of RIT Scholar Works. For more information, please contact ritscholarworks@rit.edu.

Modeling and Simulating Chemical Weapon Dispersal Patterns in DIRSIG

by

Peter Arnold

B.S. Imaging Science

Rochester Institute of Technology, Rochester NY

1997

A thesis submitted in partial fulfillment of the
Requirements for the degree of Master of Science
In the Chester F. Carlson Center for Imaging Science
Of the College of Science
Rochester Institute of Technology, Rochester NY
1999

Signature of Author: _____

Accepted by: _____

Coordinator, M.S. Degree Program

12/14/00
Date

CHESTER F. CARLSON
CENTER FOR IMAGING SCIENCE
COLLEGE OF SCIENCE
ROCHESTER INSTITUTE OF TECHNOLOGY
ROCHESTER, NEW YORK

CERTIFICATE OF APPROVAL

M.S. Degree Thesis

The M.S. Degree Thesis of Peter S. Arnold
has been examined and approved by the thesis committee
as satisfactory for the thesis requirement for the
Master of Science degree

Dr. John R. Schott, Thesis Advisor

Dr. Joseph P. Hornak

Mr. Scott D. Brown

Mr. Rolando V. Raqueño

11/11/2000

Date

THESIS RELEASE PERMISSION
ROCHESTER INSTITUTE OF TECHNOLOGY
COLLEGE OF SCIENCE
CHESTER F. CARLSON
CENTER FOR IMAGING SCIENCE

Title of Thesis:

Modeling and Simulating Chemical Weapon Dispersal Patterns in DIRSIG

I, Peter S. Arnold, hereby grant permission to Wallace Memorial Library of R.I.T. to reproduce my thesis in whole or in part. Any reproduction will not be for commercial use of profit.

Signature _____ 12/11/00

Date

Modeling and Simulating Chemical Weapon Dispersal Patterns in DIRSIG

Abstract

Fieldable thermal infrared hyperspectral imaging spectrometers have made it possible to design and construct new instruments for better detection of battlefield hazards such as chemical weapon clouds. The availability of spectroscopic measurements of these clouds can be used not only for the detection and identification of specific chemical agents but also to potentially quantify the lethality of the cloud. The simulation of chemical weapon dispersal patterns in a synthetic imaging environment offers significant benefits to sensor designers. Such an environment allows designers to easily develop trade spaces to test detection and quantification algorithms without the need for expensive and dangerous field releases.

This research focuses on the implementation of a generic gas dispersion model that has been integrated into the Digital Imaging and Remote Sensing Generation (DIRSIG) model. The gas cloud model utilizes a 3D Gaussian distribution based on theory to predict factory stack gas plumes. The model incorporates first order dynamics (drift and dispersion) to drive the macro-scale cloud development and movement. The model also attempts to account for turbulence by using fractal fractional Brownian motion techniques to reproduce the micro-scale variances within the cloud. The cloud path-length concentrations are then processed by the DIRSIG radiometry sub-model to compute the emission and transmission of the cloud body on a per-pixel basis.

Example hyperspectral image cubes containing common agents and release amounts are presented. Time lapse sequences are also provided to demonstrate the evolution of the cloud over time. Finally, recommendations and limitations of the model are listed for future improvements.

Acknowledgments

I would like to thank all those who helped, supported, put up with me, gave me ideas, and brain stormed in the evolution and finalization of this research. Dr. John R. Schott, whose suggestions and guidance made it possible for me to further my education. Mr. Scott D. Brown, who spent many hours modifying DIRSIG, explaining DIRSIG, and brain storming ideas and concepts crucial to the development of the gas cloud model. Believe me, if there ever is a next time I will use LATEX. Mr. Rolando V. Raqueño for his help with C-shell scripting and spectrally correlated noise. Dr. Joseph P. Hornak for our insightful discussions over the chemistry and physics behind gas dispersion and absorption. Ms. Julia A. Barsi for being a friend, study partner, and a person I can bounce ideas off of. Thanks for letting me finish before you! Mr. John P. Knapp for all the late night study sessions explaining Roger's homework problems. Stephanie C. VanGorden for making the DIRS lab a fun place to work. My girlfriend, Shelby, for her love and support. Without you I would have never made it! Finally, to the DIRS group, a great bunch of people. I am proud to have worked with such an insightful and helpful crew. In such a small community I am sure our paths will cross again. Take care.

PS Stay away from Microsoft WORD!!!

Dedication

No man is an island, entire of itself;
every man is a piece of the continent, a part of the main;
if a clod be washed away by the sea, Europe is the less, as well as if a promontory were,
as well as if a manor of thy friends or of thine own;
any man's death diminishes me, because I am involved in mankind;
and therefore never send to know for whom the bell tolls;
it tolls for thee.

—John Donne

This work is dedicated to my mother and father. Thank you for your love and support. I love you both.

Table of Contents

| | | |
|------------|---|-----------|
| 1 | <i>Introduction</i> | 5 |
| 2 | <i>Radiation Propagation</i> | 9 |
| 2.1 | Interaction Between Light and Matter | 9 |
| 2.1.1 | Gas Absorption | 9 |
| 2.1.2 | Thermal Self-Emission | 18 |
| 2.1.3 | Atmospheric Considerations and Simulation | 20 |
| 2.1.4 | Self-emission in the MWIR (night) and LWIR | 23 |
| 2.1.5 | Self-emission in the MWIR (day) | 25 |
| 3 | <i>Stochastic Modeling</i> | 28 |
| 3.1 | Probability Theory: Random Variables | 29 |
| 3.2 | Probability Theory: Random Fields | 30 |
| 3.3 | Probability Theory: Correlation Measures | 31 |
| 3.4 | Random Fractal Theory | 32 |
| 3.4.1 | Fractional Brownian Motion | 32 |
| 3.4.2 | Spectral Density | 33 |
| 3.5 | Fractional Brownian Motion Algorithm | 35 |
| 3.5.1 | Spectral Synthesis | 35 |
| 4 | <i>Turbulent Diffusion</i> | 38 |
| 4.1 | Conservation of Matter | 38 |
| 4.2 | Probability Density Functions: $P(x)$, $P(y)$, $P(z)$ | 41 |
| 4.3 | The Gaussian Model | 43 |
| 4.4 | Dispersion Coefficients | 44 |
| 4.4.1 | Lagrangian Statistics of Particle Motion | 44 |
| 4.4.2 | Pasquill Stability Classes and Experimental Dispersion Coefficients | 48 |
| 4.4.3 | Briggs Plume or Cloud Rise | 53 |
| 4.4.4 | Briggs Buoyancy Flux | 54 |
| 4.4.5 | Briggs Stability Parameter | 56 |
| 4.4.6 | Briggs Plume Rise Δh and Δh_{\max} for Hot, Buoyant Bent-over Plumes | 57 |
| 4.4.7 | Briggs Plume Rise Δh and Δh_{\max} for Cold, Jet Bent-over Plumes | 61 |
| 4.4.8 | Briggs Plume Rise Δh and Δh_{\max} for Vertical Plumes | 63 |
| 4.5 | Determining Cloud Dispersion and Rise Parameters | 64 |
| 5 | <i>Limitations of Gaussian Models</i> | 65 |
| 5.1 | Assumptions and Constraints | 65 |

| | | |
|-----------|---|------------|
| 6 | <i>Rendering</i> | 67 |
| 6.1 | Raytracing | 67 |
| 6.2 | 3D-DDA | 68 |
| 6.2.1 | Ray Tracing Gas Clouds in DIRSIG..... | 71 |
| 6.3 | Noise Artifacts | 78 |
| 7 | <i>Validation</i> | 81 |
| 7.1 | Voxel file validation | 81 |
| 7.2 | DIRSIG sampling | 82 |
| 7.3 | DIRSIG debug images | 84 |
| 8 | <i>DIRSIG synthetic scenes</i> | 87 |
| 8.1 | Examples | 87 |
| 9 | <i>Conclusion and Improvements</i> | 99 |
| 10 | <i>Appendix A: Voxel Creation Code</i> | 102 |
| 10.1 | Make_blob.cc and blob.dat | 102 |
| 10.2 | FBm.pro..... | 105 |
| 11 | <i>Appendix B: DIRSIG Configuration File Changes</i> | 108 |
| 11.1 | Original Configuration File Example..... | 108 |
| 11.2 | Gas Model Configuration File Example..... | 112 |
| 12 | <i>Appendix C: Creating Animations</i> | 118 |
| 13 | <i>Appendix D: Adding Spectral Noise in ENVI</i> | 120 |
| 13.1 | Adding Spectral Noise Using EigenVector-Based Transforms..... | 120 |
| 13.2 | Adding Spectrally Correlated Noise Using a Dark Current Image..... | 124 |
| 13.3 | Adding Spectrally Correlated Noise Without a Dark Current Image | 129 |
| 14 | <i>Appendix E: Design Experiment for Truth Data Via a Benign Gas</i> | 131 |
| 14.1 | Design Outline | 131 |
| 15 | <i>References</i> | 133 |

Table of Figures

| | |
|---|----|
| <i>Figure 2-1 Characteristics of absorption spectra [Schott, 1997]</i> | 10 |
| <i>Figure 2-2 Computation of the absorption coefficient [Schott, 1997]</i> | 12 |
| <i>Figure 2-3 Toxicity of nerve agents [ICA, 1997]</i> | 16 |
| <i>Figure 2-4 Toxicity of fire gases [Babrauskas, 1997]</i> | 17 |
| <i>Figure 2-5 Atmospheric transmission, solar irradiance, and earth self-emission spectra [Schott, 1997]</i> | 21 |
| <i>Figure 2-6 Sources of thermal radiation</i> | 23 |
| <i>Figure 2-7 Relationship between terms in the "Big Equation" and energy paths associated with the radiance reaching the sensor [Schott, 1997]</i> | 26 |
| <i>Figure 3-1 Probability of observations for a normal distribution</i> | 30 |
| <i>Figure 3-2 Gaussian 2D & 3D distributions, without texture map</i> | 36 |
| <i>Figure 3-3 Fractional Brownian motion 2D & 3D texture maps</i> | 37 |
| <i>Figure 3-4 Gaussian*Fbm 2D & 3D distributions</i> | 37 |
| <i>Figure 4-1 Instantaneous point sources spaced at time interval, dt, used to create a continuous source</i> | 40 |
| <i>Figure 4-2 1D Gaussian distribution function</i> | 43 |
| <i>Figure 4-3 Particle motion from insantaneous source</i> | 45 |
| <i>Figure 4-4 Taylor weighting function of energy spectrum that causes diffusion as a function of time T [Blackadar, 1997]</i> | 47 |
| <i>Figure 4-5 Pasquill Stability Types [Beychok, 1994]</i> | 49 |
| <i>Figure 4-6 McMullen Equation Constants for Rural Diffusion [Beychok, 1994]</i> | 51 |
| <i>Figure 4-7 Gifford Equation Constants for Urban Diffusion [Beychok, 1994]</i> | 52 |
| <i>Figure 4-8 Ambient Temperature Gradient</i> | 57 |
| <i>Figure 6-1 Simple Raytracing Process [Fusner, 1999]</i> | 67 |
| <i>Figure 6-2 3D-DDA</i> | 70 |
| <i>Figure 6-3 Ray tracing a gas cloud</i> | 71 |
| <i>Figure 6-4 Transmission: base 10 vs base e</i> | 75 |
| <i>Figure 6-5 CLT vs Step Wise calculations for cloud radiance, part 1 of 3</i> | 76 |
| <i>Figure 6-6 CLT vs Step Wise calculations for cloud radiance, part 2 of 3</i> | 77 |
| <i>Figure 6-7 CLT vs Step Wise calculations for cloud radiance, part 3 of 3</i> | 77 |
| <i>Figure 6-8 Diagram of step wise 1D group of cells</i> | 78 |

| | |
|--|-----|
| Figure 6-9 Noise and information degrading effects in a remote sensing system [Kerekes, 1987]..... | 79 |
| Figure 7-1 Geometric construction for sampling transformation..... | 83 |
| Figure 8-1 X,Y scale in meters..... | 88 |
| Figure 8-2 Desert animation sequence..... | 88 |
| Figure 8-3 Point A, cloud emission with background interaction..... | 89 |
| Figure 8-4 Point B, Gas cloud absorbing against atmosphere | 89 |
| Figure 8-5 Point B2, Gas cloud emission against background..... | 90 |
| Figure 8-6 Point C, high cloud emission..... | 90 |
| Figure 8-7 Point D, background spectra..... | 91 |
| Figure 8-8 Point E, atmospheric spectra..... | 91 |
| Figure 8-9 X,Y image scale in meters..... | 92 |
| Figure 8-10 Desert release, 1500 seconds | 92 |
| Figure 8-11 X,Y scale in meters..... | 93 |
| Figure 8-12 Urban animation sequence..... | 94 |
| Figure 8-13 Spectral "finger print" radiance for center of GD gas cloud..... | 95 |
| Figure 8-14 Spectral "finger print" radiance for edge of GD gas cloud..... | 95 |
| Figure 8-15 Absorption coefficients of Soman (GD) 800 - 1050 [cm^{-1}]..... | 96 |
| Figure 8-16 Soman (GD) gas cloud transmission for Figure 8-13 and Figure 8-14 | 96 |
| Figure 8-17 Soman 20 kg release background: original, zoom, and spectral profile for location indicated by crosshairs..... | 97 |
| Figure 8-18 Soman 20 kg release horizon: original, zoom, and spectral profile for location indicated by crosshairs (taken just above horizon)..... | 97 |
| Figure 8-19 Soman 2.5 kg release background: original, zoom, spectral profile for location indicated by crosshairs | 98 |
| Figure 8-20 Soman 2.5 kg release horizon: original, zoom, spectral profile for location indicated by crosshairs (taken just above horizon)..... | 98 |
| Figure 13-1 Increased seperability and variance due to the projection onto a rotated principle component axis [Schott, 1997] | 121 |

1 Introduction

The capability to digitally model and simulate chemical weapon (CW) clouds is crucial in the development of detection methods designed to keep military and civilian personnel well away from chemical weapon release. Both the release and modeling of CW clouds covers a broad range of complex phenomena. The manner in which the CW is released, the terrain over which it travels, surrounding buildings, initial momentum, seasonal variations, possible canopy structure, atmospheric pressure, environmental parameters such as solar insolation, rain, and wind speed all effect the evolution of the gas cloud. This research concentrated only on a small portion of the much larger picture. The main goal was to provide a simple model that predicts the dispersion, advection, dissipation, concentration, and temperature with respect to time of these deadly clouds. Potential improvements to the model will be discussed in the section titled: Conclusion and Improvements.

The CWs of interest in this research are known as nerve agents. All nerve agents in their pure state are colorless liquids at STP. Nerve agents inhibit the release of the enzyme acetylcholinesterase. This affects nerve impulses causing severe muscle cramping and death by suffocation. All nerve agents belong to the chemical group of organo-phosphorus compounds. They are stable and easily dispersed, highly toxic and have rapid effects (2-30 minutes) when absorbed though the skin and via respiration. [ICA, 1997]

Nerve agents are typically disseminated in three manners. The first is to use a burster or binary technology (3-7 kg). Two liquid components are separated in a projectile by a rupture disc.

When the projectile is fired the disc bursts and the two components combine. Rifling in the barrel gives the projectile a spinning velocity that mixes the nerve agent. The second method is to mix the components before launch and store them in a ballistic missile warhead (200-1000 kg). The third is to let the agent leak out and be dispersed by the air stream. This research concentrates on discrete sources and is applicable mainly to binary and warhead dissemination. However, continuous sources can be treated as discrete sources that are closely placed together. Therefore any number of continuous source plume models could be used to model the third case scenario.

Spectral data was supplied through the Army's Aberdeen Proving Grounds for the following nerve agents: VX, Distilled Mustard (HD), Soman (GD), Sarin (GB), and Tabun (GA). The spectral features for these nerve agents are in the mid-wave (MWIR) to long wave infrared (LWIR), 4-24 μ m. In this region the primary sources of radiance reaching the sensor are thermal self-emission, upwelled thermal emission, background thermal emission, and earth thermal emission. The self-emission of the gas is dependent on the temperature and emissivity of the gas. The transmission of the gas will be dependent on the spectral absorbance of the nerve agent. Scattering is ignored since it is considered negligible above 2.5 μ m. [Kuo, 1997]

In this research, synthetic image generation (SIG) was used to simulate the CW clouds and investigate remote sensing capabilities in detection. The simulations were run using Soman (GD) although any of the nerve agents could have been used. SIG uses first principles from physics to produce radiometrically accurate images as seen by a sensor. SIG allows the user to vary scene and sensor geometries, wavelengths, meteorological conditions, background interactions, and atmospheric profiles without the need for dangerous and expensive field releases.

The rendering of these synthetic scenes will be done with the Digital Imaging and Remote Sensing Image Generation (DIRSIG) code. DIRSIG is a ray tracing code developed by the Digital Imaging and Remote Sensing lab (DIRS) at the Center for Imaging Science at the Rochester Institute of Technology in Rochester, NY. It creates radiometrically accurate synthetic images for various sensor platforms [Brown, 1999]¹. DIRSIG incorporates the Moderate Resolution Transmittance (MODTRAN) code [Berk, 1989] to model the atmosphere. To support the higher spectral resolution needed on this effort, the Fast Atmospheric Signature Code (FASCODE) [Smith, 1978] was added as a source of atmospheric parameters. The points of contact, general descriptions, documentation references, and software downloads for both MODTRAN and FASCODE are available at <http://www-vsbn.plh.af.mil>. The THERM [DCS Corporation, 1990] thermal sub-model is incorporated in DIRSIG to predict time dependent temperatures of objects within the scene as influenced by their environment.

The simulation of a time sequence depicting the evolution of a CW cloud is quite complex and requires modeling capabilities that are still evolving. This type of simulation is known as physical dynamic modeling. The user does not specify the entire phenomenon, but rather provides external forces and material properties. The model then predicts the evolution of the position and shape of the gas over time based on physics. The model presented in this research draws from theory used in the morphology of smoke plumes [Beychok, 1994 & Blackadar, 1997]. The forms that smoke plumes exhibit are a result of diffusion and atmospheric turbulence due to heating currents and wind deviations. Knowledge based on measurements acquired over a long period have been used to successfully predict smoke plume movement. Traditionally these smoke plume models use Fickian Diffusion that predicts a concentration with a Gaussian or Normal probability distribution function (PDF). The Fick equation states that the rate of change of the concentration of a property depends on the divergence of the three-dimensional flux of the

¹ Download available from <http://www.cis.rit.edu/~dirsig/doc/index.html>

property, and that in an inhomogeneous environment the flux is proportional to the gradient of the mean concentration. The Gaussian shape of predicted plumes and clouds is consistent with most experimental data if sufficient allowance is given to sampling irregularities [Blackadar, 1997]. This technique describes the macroscopic properties or global shape of the gas cloud. A fractional Brownian motion (fBm) field is then introduced to add turbulent small-scale detail. Fractional Brownian motion can be used to describe the irregular thermal motion of the gas molecules and fits under the concept of fractal geometry [Crownover, 1995].

At this time a blast model is not available. A blast model would help in developing the initial kinetic theory describing the evolution of the gas until some equilibrium was reached at some time, $t_2 > 0$. The model presented in this research describes the evolution after reaching equilibrium, t_2 .

2 Radiation Propagation

This section discusses the basic physics involved with quantifying the radiance from target to sensor. This covers the interaction between light and matter and the radiometric equations involved with the remote sensing of gas clouds.

2.1 Interaction Between Light and Matter

2.1.1 Gas Absorption

Different gasses attenuate light at various wavelengths. These absorption features can be used as "finger prints" which can identify the gas. The field of identifying these "finger prints" is known as spectroscopy. The intensity and shape of these lines are a function of temperature, molecular weight of the gas concentration, and relative pressure. One important spectral broadening process is caused by the Doppler effect, in which radiation is shifted in frequency when the source is moving towards or away from the observer. Doppler theory states that frequency increases with temperature and decreases with molecular mass according to:

$$\Delta v \propto \sqrt{\frac{T}{MW}}$$

Equation 2-1

where Δv = width of absorption frequency, T = Temperature [Kelvin], MW = gram molecular weight [g/mol]

The degree of broadening is also proportional to the relative pressure [Schott, 1997]. In the MWIR these absorption lines are due to transitions in the vibrational state of the molecule. Above 20 μm rotational transitions are the dominant process. Figure 2-1 illustrates an idealized absorption spectra. Figure 2-1b demonstrates the effects of broadening caused by temperature, pressure, and molecular weight. Figure 2-1c represents the net effect of overlap between the broadened spectra. Thus, discrete absorption features can be lost due to overlap caused by broadening.

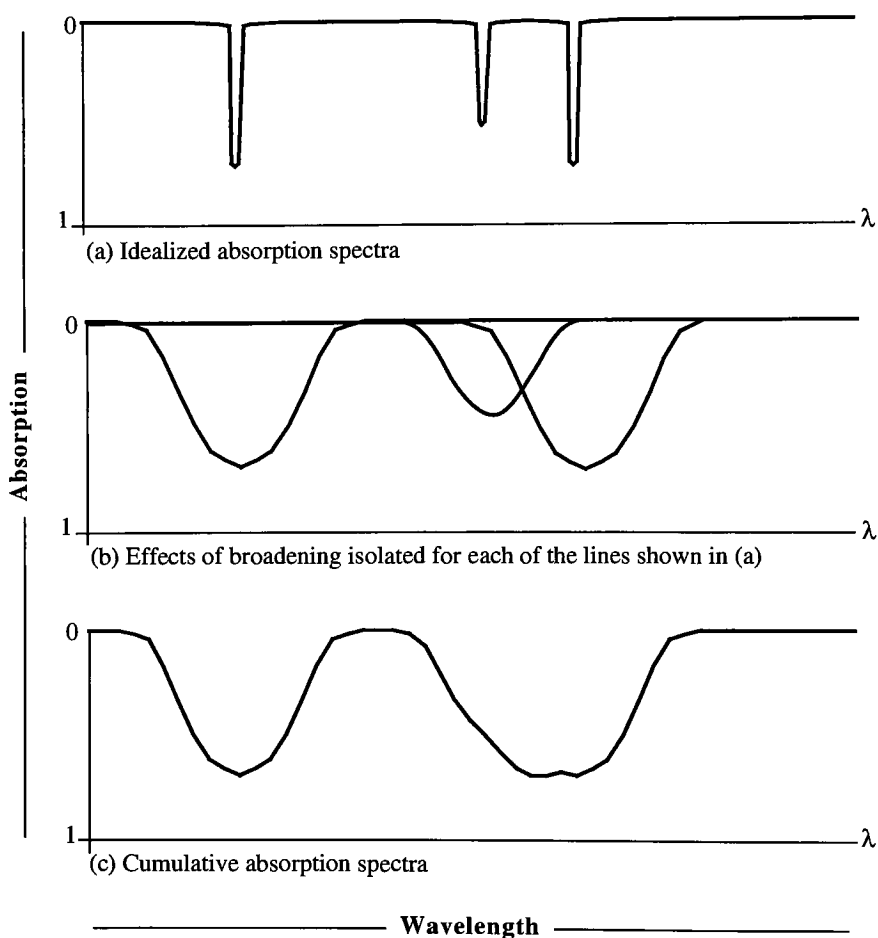


Figure 2-1 Characteristics of absorption spectra [Schott, 1997]

For a more rigorous treatment of absorption and remote sensing see Goody (1989).

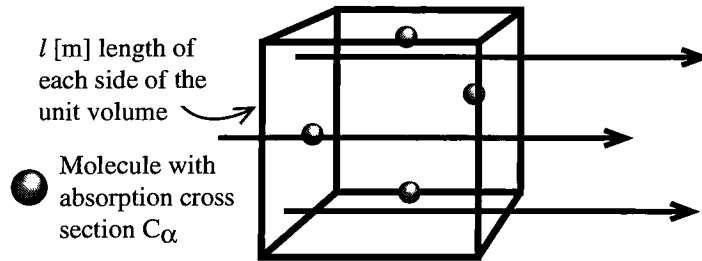
The following is from Schott [1997]:

In simplified form, we can derive the following relationships between the number and efficiency of absorbers and their effect on the propagating beam. At each wavelength, we define the absorption cross section C_α to be the effective size of a molecule relative to the photon flux at that wavelength. Conceptually, this can be expressed as:

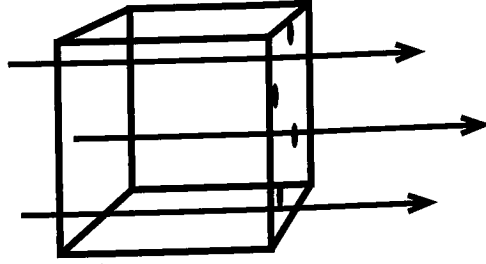
$$C_\alpha = C_g \xi = \pi r^2 \xi [\text{m}^2]$$

Equation 2-2

where $C_g [\text{m}^2]$ is the geometric cross section for a molecule of radius $r [\text{m}]$ and ξ is a unitless wavelength-dependent efficiency factor that is proportional to the molecule's ability to absorb flux. Values of C_α can be derived for particular temperatures and pressures from experimental data or through molecular energy theory, then adjusted for the effects of the temperature and pressure. The molecule is then assumed to be a perfect absorber over that cross-sectional area. To compute the fractional amount of energy lost per unit length of transit in a propagating beam, we need to know the number density of the molecules. Referring to Figure 2-2, we let m' be the number of molecules in a unit volume of side dimension $l [\text{m}]$.



(a) A unit volume containing m' absorption centers. We assume that the medium has a large mean free path such that in a small volume, if we project the molecules onto one face, there will be no overlap, i.e.,



) Projection of absorbers onto the face of the volume

Figure 2-2 Computation of the absorption coefficient [Schott, 1997]

Then, the area blocked (A_b) by the molecules is

$$A_b = m' C_\alpha [\text{m}^2]$$

Equation 2-3

the area on the face of the volume (A_f) onto which the molecules were projected

$$A_f = l^2 [\text{m}^2]$$

Equation 2-4

the fraction of the face blocked by the absorbing molecules (F) is

$$F = \frac{m' C_\alpha}{l^2} \left[\frac{\text{m}^2}{\text{m}^2} \right]$$

Equation 2-5

Therefore, the fraction amount of flux absorbed β_α per unit length of transit (l) is

$$\beta_\alpha = \frac{F}{l} = \frac{m'}{l^3} C_\alpha [\text{m}^{-1}] = \frac{m'}{V} C_\alpha = m C_\alpha [\text{m}^{-1}]$$

Equation 2-6

where V is the unit volume, m' is the number density of molecules, defined as the number of molecules per unit volume and β_α is the absorption coefficient, defined

as the fractional amount of flux lost to absorption per unit length of transit in a propagating beam.

According to Grum (1979), for an element of path length $dz[m]$ in the medium, the element of fractional flux lost can be expressed as:

$$\frac{d\Phi}{\Phi} = -\beta_{\alpha}(z)dz$$

Equation 2-7

where we have made the dependence of β_{α} on location in the media explicit. For propagation along a finite path starting at distance zero where we have initial flux Φ_0 to distance z where we have flux Φ_z , we have

$$\begin{aligned} \int_{\Phi_0}^{\Phi_z} \frac{d\Phi}{\Phi} &= \int_0^z -\beta_{\alpha}(z)dz = \ln \Phi \Big|_{\Phi_0}^{\Phi_z} = \int_0^z -\beta_{\alpha}(z)dz = \ln \Phi_z - \ln \Phi_0 \\ &= \ln \left(\frac{\Phi_z}{\Phi_0} \right) = \int_0^z -\beta_{\alpha}(z)dz \end{aligned}$$

Equation 2-8

Making both sides powers of e to simplify the left-hand side yields

$$\frac{\Phi_z}{\Phi_0} = e^{-\int_0^z \beta_{\alpha}(z)dz}$$

Equation 2-9

Recognizing the left-hand side as a definition of transmission (ratio of flux out to flux in) and solving for the simplified case of a homogeneous medium, we have

$$\tau = \frac{\Phi_z}{\Phi_0} = e^{-\beta_{\alpha} \int_0^z dz} = e^{-\beta_{\alpha} z}$$

Equation 2-10

which is variously known as Lambert's law or Bouguer's law.

The product $\beta\alpha z$ is generally referred to as the *optical depth* (δ), i.e.,

$$\delta_a = \beta\alpha z$$

Equation 2-11

To this point we have implicitly assumed a media containing a single constituent. For a homogeneous media containing many types of molecules, we introduce the subscript i to denote the particular constituent. If we assume that the molecules interact independently with the propagating flux, we can express the transmission as:

$$\tau = \prod \tau_i = e^{-\sum \delta_i} = e^{-\sum \beta_{ai} z} = e^{-\sum m_i C_{ai} z} = e^{-\beta_a z} = e^{-\delta_a}$$

Equation 2-12

where Π designates the product of the transmission values for each constituent if computed separately, the summation (Σ) is over all constituents, and we redefine $\beta\alpha = \Sigma \beta_{ai}$ to be the composite absorption coefficient and δ_i to be the composite optical depth due to absorption. [Schott, 1997]

Finally the relationship between absorption and transmission is expressed as:

$$A = -\ln(\tau)$$

Equation 2-13

The spectral absorption "finger print" is a function of the absorption coefficient, concentration of the gas, and the path-length over which the absorption was measured. The raw data was provided through the Army's Aberdeen Proving Grounds, MD. For published data see Hoffland (1985). The raw data provided units of absorption coefficients in liters/(gram*cm), concentration in grams/liter, and path-length in meters. To determine absorption from Bouguer's Law:

$$A = \beta_{\alpha} C z$$

Equation 2-14

Where β_{α} = absorption coefficient [L/g*cm], C = concentration [g/L], and z = path-length of the cell [m]

The concentration of gases is often given in parts per million [ppm] and is known as the volume-mixing ratio (VMR). The definition of 1 ppm of a gas means there is one part of gas per 1 million parts of air. The VMR can be computed through the following derivation:

$$\text{molar density [mol/L]} = \frac{\text{concentration [g/L]}}{\text{molecular weight [g/mol]}}$$

Equation 2-15

$$\text{molar density } \left[\frac{\text{mol}}{\text{m}^3} \right] = \text{molar density } \left[\frac{\text{mol}}{\text{L}} \right] * \frac{1000[\text{L}]}{1[\text{m}^3]}$$

Equation 2-16

$$\text{VMR [ppm]} = \text{molar density } \left[\frac{\text{mol}}{\text{m}^3} \right] * 0.0224 \text{ volume of ideal gas @ STP } \left[\frac{\text{m}^3}{\text{mol}} \right] * 1000000$$

Equation 2-17

A similar conversion is needed to convert the absorption coefficient to 1/ppm-m:

$$\beta_{\alpha} \left[\frac{1}{\text{ppm} \cdot \text{m}} \right] = \frac{\beta_{\alpha} \left[\frac{\text{L}}{\text{g} \cdot \text{cm}} \right] * \text{molecular wt} \left[\frac{\text{g}}{\text{mol}} \right] * \frac{1[\text{m}^3]}{1000[\text{L}]} * \frac{100[\text{cm}]}{1[\text{m}]}}{0.0224, \text{ normal gas volume @ STP } \left[\frac{\text{m}^3}{\text{mol}} \right] * 1000000}$$

Equation 2-18

STP stands for standard temperature (273.15 K) and pressure (1 ATM). The gas volume can be adjusted for various temperatures and pressures by using the ideal gas law:

$$\text{Gas Volume } \left[\frac{\text{m}^3}{\text{mol}} \right] = \frac{R \left[\frac{\text{L} \cdot \text{atm}}{\text{mol} \cdot \text{K}} \right] \cdot T[\text{K}]}{P [\text{atm}]} \cdot \frac{[\text{m}^3]}{1000[\text{L}]}$$

Equation 2-19

Where R = Universal gas constant = kN_a = Boltzmann's constant * Avogadro's number = $8.2057\text{e-}2$ [L*atm/mol*K], T = temperature [K], and P = pressure [atm].

The following table illustrates lethal dosages of 4 common nerve agents. This should help the reader understand the extreme toxicity of these substances. For gases, the toxicity is expressed by LC_{t50} , where L stands for lethal, C stands for concentration, t stands for time, and 50 means 50% of the exposed population will die due to their injuries. Thus, Tabun vapor at 200 [mg/m³] for 1 minute has the same toxicity as a 100 [mg/m³] for two minutes. For liquid exposure, the toxicity is expressed by LD_{50} , where D stands for dosage and all other symbols are the same.

| Agent | LC_{t50} [mg*min/m ³] | LC_{t50} [ppm*min] | LD_{50} [mg] |
|-----------|--|-------------------------|----------------|
| Tabun, GA | 200 | 27.64 | 4000 |
| Sarin, GB | 100 | 15.99 | 1700 |
| Soman, GD | 100 | 12.29 | 300 |
| VX | 50 | 4.19 | 10 |

Figure 2-3 Toxicity of nerve agents [ICA, 1997]

By contrast the following table illustrates the toxicity of some primary gases found in fires. The reader should be aware that for fire toxicity data the exposure period normally used is 30 minutes. If the biological effect is linear (Haber's Law) then these numbers would be multiplied by 30 to give LC_{t50} [ppm*min]. I.e., The lethal concentration of carbon monoxide for an exposure of 1 minute would be $3000 \text{ [ppm]} * 30 \text{ [min]} = 90000 \text{ [ppm*min]}$. Almost all gases show deviations from this simple relationship, but for rough estimating purposes it can be used if better data is unavailable. [Babrauskas, 1997]

| Gas | LC_{t50} [ppm*30 min] |
|-------------------|-------------------------|
| Carbon Monoxide | 3000 |
| Ammonia | 9000 |
| Hydrogen Chloride | 3700 |
| Acetic Acid | 11000 |

Figure 2-4 Toxicity of fire gases [Babrauskas, 1997]

2.1.2 Thermal Self-Emission

The emissivity characterizes the radiating efficiency of a surface and is material dependent. It is a unitless number with range 0 -> 1, with 1 being a perfect emitter. According to the second law of thermodynamics the rate of emission at a given wavelength must equal the rate of absorption at that wavelength. Thus, all of the incident radiation on the ideal emitter would be absorbed. Since no radiation is reflected from the surface an ideal emitter is often called a blackbody.

The spectral radiant exitance from a blackbody has the following three general features:

1. At a given temperature, the Stefan-Boltzmann Equation gives the total exitance:

$$M \left[\frac{W}{m^2} \right] = \sigma \left[\frac{W}{m^2 K^4} \right] * T^4 [K]$$

Equation 2-20

where σ = Stefan-Boltzmann constant = $5.67e-8$ [W/(m²K⁴)] and T = [Kelvin]

2. If the temperature is increased, the spectral radiant exitance increases for every wavelength.
3. The spectral exitance has a single peak that shifts to smaller wavelengths as temperature increases:

$$\lambda_{\max} = \frac{A[\mu m K]}{T[K]}$$

Equation 2-21

Where A = Wien displacement constant = 2898 [μ m] and T = temp [K]

As a function of wavelength the Planck Equation expresses the spectral radiant exitance from a blackbody:

$$M(\lambda)_{bb} = \frac{2\pi hc^2}{\lambda^5 (e^{\frac{hc}{\lambda T}} - 1)} [W/(m^2 \mu m)]$$

Equation 2-22

Where λ = wavelength [μm], h = Planck's constant $6.6256e-34$ [joules sec], c = speed of light $3e8$ [m/s], T = temp [K], K = Boltzmann gas constant $1.38e-23$ [joules/K]

The Planck equation can be expressed in terms of spectral radiance for Lambertian surfaces as:

$$L(\lambda) = \frac{M(\lambda)}{\pi} \left[\frac{W}{m^2 sr \mu m} \right]$$

Equation 2-23

The emissivity is then defined as:

$$\varepsilon(\lambda) = \frac{M(\lambda, T)}{M_{bb}(\lambda, T)}, \quad 0 < \varepsilon(\lambda) < 1$$

Equation 2-24

Where $M(\lambda, T)$ = radiant exitance of an object at a given wavelength and temperature and

$M_{bb}(\lambda, T)$ = radiant exitance of a blackbody at the same wavelength and temperature.

For a gas volume the emissivity can be expressed as a function of the transmission loss due to absorption as:

$$\varepsilon(\lambda) = 1 - \tau(\lambda) = 1 - e^{-K_{\text{abs}}z}$$

Equation 2-25

Where $\tau(\lambda)$ = transmittance and $\varepsilon(\lambda)$ = emissivity

2.1.3 Atmospheric Considerations and Simulation

Atmospheric absorption results in the loss of radiant energy to atmospheric components. In the LWIR the main absorption constituents are H₂O, O₃, and CO₂. Areas where the atmospheric transmission is high are known as atmospheric windows. In thermal sensing the atmospheric windows exist at 3-5 μ m and 8-14 μ m. In the 8-14 μ m region solar photon contribution is lost in the noise limits and can be ignored. In the 3-5 μ m region solar photon and thermal photon contribution are of the same order of magnitude. Therefore, in the 3-5 μ m region under daylight conditions both solar and thermal photons must be considered.

Figure 2-5 illustrates solar vs. thermal photon contribution with respect to atmospheric transmission across the visible and infrared spectrum.

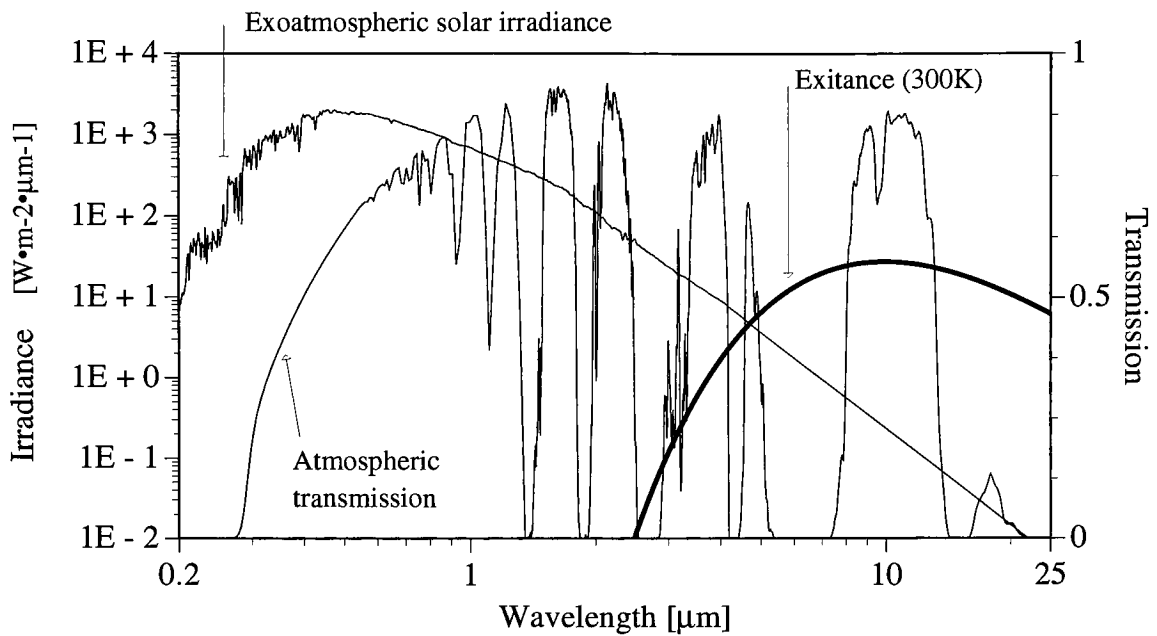


Figure 2-5 Atmospheric transmission, solar irradiance, and earth self-emission spectra [Schott, 1997]

Figure 2-5 shows the selection of a sensor must consider the atmospheric window, spectral sensitivity of the sensor, along with source, magnitude, and spectral composition of the photons available [Lillesand, 1994]. For the work reported here, solar and self emissive calculations are included at all wavelengths, however, scattering effects in the gas plume are not included. Scattering is typically insignificant in the MWIR and LWIR and should not introduce any significant errors when the cloud droplets are small compared with the imaging wavelength.

The atmosphere will be modeled using the atmospheric transport code MODTRAN for delta wave number increments > 2 $[\text{cm}^{-1}]$. This program is a computationally rigorous radiation transfer algorithm that models the spectral absorption, transmission, emission, and scattering characteristics of the atmosphere. MODTRAN assumes that the atmosphere is a set of homogeneous layers. The characteristics of these layers are either modeled by several default

model atmospheres or it can be characterized by radiosonde data collected for a specific atmosphere. In the regions of interest for this research (8-14 μm and 3-5 μm at night) where solar photons are ignored, the radiance reaching the sensor can be approximated by:

$$L_{\lambda} = \varepsilon L_{T\lambda} \prod_{i=1}^N \tau_{i\lambda} + \sum_{i=1}^N \left[(1 - \tau_{i\lambda}) L_{T_i\lambda} \prod_{j=i+1}^N \tau_{j\lambda} \right]$$

Equation 2-26

Where $\tau_{i\lambda}$ = transmission through the i th layer along a path length z_i at wavelength λ , $L_{T_i\lambda}$ = blackbody spectral radiance with temperature T_i of the i th layer.

High resolution runs, for wave numbers < 2 [cm^{-1}], used FASCODE. This model is a line-based method versus MODTRAN band-based method.

The next two sections cover the thermal governing equations for remote sensing of gaseous clouds.

2.1.4 Self-emission in the MWIR (night) and LWIR

Thermal sources are the significant contributors in the MWIR (at night) and LWIR. The radiance reaching the sensor is a function of Planck's Equation, Equation 2-22, modified by the emissivity as a function of wavelength, defined as one minus the transmission, Equation 2-25. The primary sources of interest are: gas cloud self-emission ($L_{S\varepsilon}$), upwelled thermal radiance ($L_{U\varepsilon}$), reflected downwelled thermal radiance ($L_{D\varepsilon}$), reflected background ($L_{B\varepsilon}$) and earth thermal radiance ($L_{E\varepsilon}$).

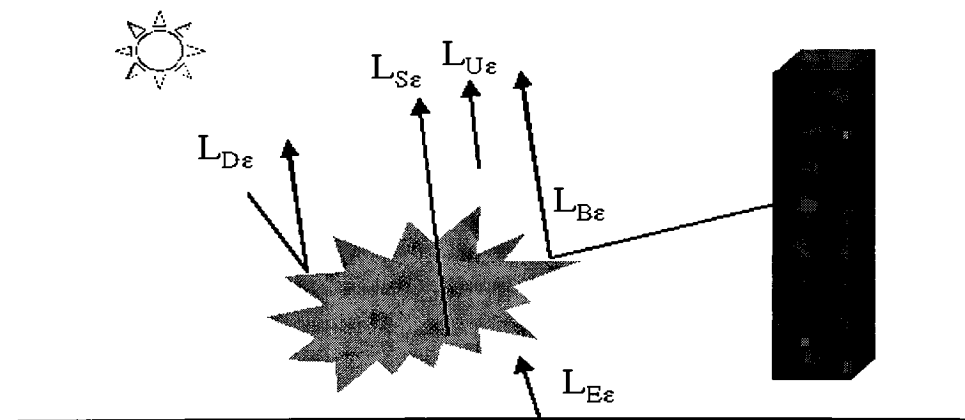


Figure 2-6 Sources of thermal radiation

The thermal radiance reaching the sensor without the gas cloud can be expressed as:

$$L = L_{E\epsilon} + L_{D\epsilon} + L_{B\epsilon} + L_{U\epsilon}$$

$$= \left[\epsilon L_{E\epsilon} + (F E_{D\epsilon} + (1 - F) E_{B\epsilon}) \frac{r_d}{\pi} \right] \tau_2 + L_{U\epsilon}$$

Equation 2-27

Where r_d = diffuse reflectance (constant at all angles), τ_2 = transmission from target to sensor, F = shape factor, and E = irradiant [W/m^2]

The earth self-emission term ($E\epsilon$) dominates this expression with significant contribution from upwelled radiance ($U\epsilon$). The reflected downwelled radiance ($D\epsilon$) and reflected background radiance ($B\epsilon$) terms are typically much smaller, though still significant contributors if measurement accuracy's of tenths of a degree are desired. The relative importance of these reflected terms will decrease with increasing emissivity (decreasing reflectivity), but in general they will not be negligible until emissivity values approach 0.99. The relative importance of the downwelled radiance and the background radiance is controlled by the shape factor (F). The shape factor represents the fraction of hemisphere above the target that is sky. For nearly horizontal unobstructed surfaces, the shape factor F approaches 1.0 and the background term becomes negligible. [Schott, 1997]

The temperature of the gas cloud will determine the blackbody exitance as expressed by Planck's Equation, Equation 2-22. The blackbody exitance multiplied by the emissivity as a function of λ determines the gas cloud self-emission. In simple form this can be approximated as:

$$L_{C\epsilon} = (1 - \tau_g) L_{T_g}$$

Equation 2-28

Where τ_g = effective transmittance of the gas and L_{T_g} = radiance due to the temperature of the gas [$\text{W}/(\text{m}^2 \cdot \text{sr})$]

Equation 2-27 with the gas cloud then becomes:

$$L = \left[\epsilon L_{E\epsilon} + (F E_{D\epsilon} + (1 - F) E_{B\epsilon}) \frac{\tau_d}{\pi} \right] \tau_2 \tau_g + (1 - \tau_g) L_{Tg} + L_{U\epsilon}$$

Equation 2-29

2.1.5 Self-emission in the MWIR (day)

The MWIR has a significant solar and thermal contribution. Hence both solar and thermal radiance must be included when solving for total radiance. Figure 2-7 includes the “Big Equation” which calculates the solar and thermal radiance reaching the sensor.

Figure 2-7 illustrates the total radiance reaching the sensor from a target on the ground for both solar and thermal photons. Path A is the exoatmospheric solar radiance, Path B is skylight or downwelled radiance, Path C is upwelled solar radiance, Path G is background solar radiance, Path D is thermal self-emission, Path E is downwelled thermal emission, Path F is upwelled thermal emission, and Path H is background thermal emission.

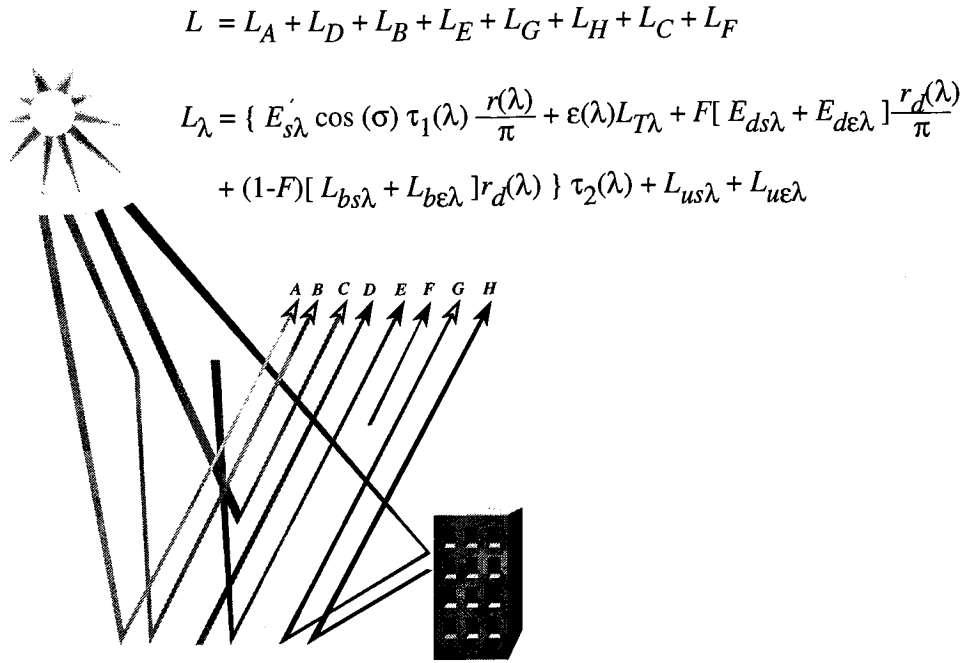


Figure 2-7 Relationship between terms in the "Big Equation" and energy paths associated with the radiance reaching the sensor [Schott, 1997]

All paths in Figure 2-7 are included in the DIRSIG simulations. In addition a gas cloud will attenuate both thermal and solar photons and emit thermal photons as discussed in section 6.2.1. Because the focus is on the MWIR and LWIR where scattering is typically low, scattering effects from the gas cloud are not included. This should not introduce any significant errors when the cloud is very gaseous, i.e. gas droplets are small relative to imaging wavelength. However, if a significant number of droplets or large particles are associated with the cloud then scattering should be considered for future upgrades.

The previous sections lay down the physics (the "Big equation") involved with quantifying the energy emitted and absorbed by a gas cloud. They also discuss the energy contributions from background and atmospheric interactions. The sections propose ways to calculate and quantify various sources in order to model what the sensor will detect as they relate to certain atmospheric

window constraints. The reader should keep in mind that DIRSIG and MODTRAN or FASCODE are used as the primary tools to model the physics discussed in the previous sections.

3 Stochastic Modeling

Convective atmospheric transport and diffusion effects on aerosol concentrations in plumes can be represented with a Gaussian distribution. This Gaussian model visualizes aerosol plumes as smooth distributions resulting from time-averaged contributions of turbulence to the mean flow. Any observer of aerosol plumes knows that the distribution is not perfectly Gaussian but has some random behavior. The behavior of a probabilistic system cannot be predicted exactly but the probability of certain behaviors is known. Fractals and mathematical chaos lay the foundation for numerical techniques that can be used to generate spatially correlated random fields. The fluctuations that occur with chaos sets are only seemingly random. This pseudo-randomness spawns from a sensitive dependence on initial conditions. The input parameters introduce some error or randomness but the overall process follows some deterministic outcome.

Fractal statistics are used to generate superpositions of random fluctuations over different time scales. This randomness can be used to simulate turbulence. As pointed out by Sakas (1993); Mandelbrot (1975) and Lovejoy (1985) proposed that static images of turbulent fields can be regarded as fractals with a Hurst exponent of 0.7. This corresponds closely to experimental values of actual turbulence measured by Sreenivasan (1991).

The following sections attempt to describe a family of random fractal functions known as fractional Brownian motion (fBm). fBm has been used in many applications ranging from physical sciences and engineering to artistic applications. For a more in-depth discussion of fractals, chaos, and fBm see Thompson (1989), Crownover (1995), and Yaglom (1986). The following heuristic arguments are taken from Stam (1991 & 1995) and Peitgen (1988).

3.1 Probability Theory: Random Variables

The set of all possible outcomes for a given observation is called the sample space. A random variable X or in general the variate X is a variable that can take on any value in the sample space. The overall behavior of a random variate X can be described by its probability distribution function (PDF). The PDF is a function $P[X=x]$ meaning "the probability of variate X equal to x " defined by:

$$\text{pdf} = \int_{-\infty}^{\infty} f(x)dx = 1, \quad 0 \leq f(x) \leq 1$$

Equation 3-1

In practice, the statistics of the random variate X are used to provide useful information. These statistics include the expectation or mean, standard deviation, variance, and correlation. The mean is defined as:

$$\mu = E[x] = \int_{-\infty}^{\infty} xf(x)dx$$

Equation 3-2

The variance is a measure of how the values X are distributed around the mean:

$$\sigma^2 = \text{Var}[x] = \int_{-\infty}^{\infty} (x - \mu)^2 f(x)dx = E[(x - \mu)^2] = E[x^2] - \mu^2$$

The standard deviation σ is equal to the positive square root of the variance. It is a linear measure of how the values X are distributed around the mean. For selected values of $N\sigma$ the following probabilities for normally distributed observations are obtained:

| $N\sigma$ | 1σ | 2σ | 3σ |
|--|-----------|-----------|-----------|
| $P(\mu - N\sigma < X < \mu + N\sigma)$ | 0.6826 | 0.9544 | 0.9974 |

Figure 3-1 Probability of observations for a normal distribution

Figure 3-1 indicates that about 68 percent of normally distributed observations are between $\mu - \sigma$ and $\mu + \sigma$; about 95 percent are in the interval given by $\mu \pm 2\sigma$ and almost all are within three standard deviations of the mean μ .

3.2 Probability Theory: Random Fields

A random field is a random process that returns a random variate as a function of some discrete variable in 2 or more dimensions. A dynamic (time driven) 3D gas cloud is an example of a random 4D field. A gas cloud represents a 3D-density map and for each fixed time, $t > 0$, an instance of the previous 3D random field is observed. This random field can be denoted by $R(t)$ with spatial dimensions x, y, z as a function of time, t . This introduces $\mu = \mu(t)$ and $\sigma^2 = \sigma^2(t)$.

3.3 Probability Theory: Correlation Measures

A random field that for each value of time, $t > 0$, returns an independent random variable is known as white noise. Independence means that a value at $t_1 = 0$ has no effect on a value at $t_2 > 0$. This type of random field has no correlation from value to value. The correlation measure shows how the values of the random field R at two given positions t_1 and t_2 are related. There are three common statistical measures of correlation: variance, covariance, and normalized covariance. The variance is the mean square difference of the random field at time t_1 and t_2 :

$$\gamma(t_1, t_2) = \frac{1}{2} E[(R(t_1) - R(t_2))^2]$$

Equation 3-4

The covariance is:

$$\text{Cov}(t_1, t_2) = E[R(t_1)R(t_2)] - \mu(t_1)\mu(t_2)$$

Equation 3-5

Positive values of the covariance indicate values of the random field tend to be close. Negative values indicate a large difference in values. The normalized covariance or correlation function is

$$\text{Cor}(t_1, t_2) = \frac{\text{Cov}(t_1, t_2)}{\sigma(t_1)\sigma(t_2)}$$

Equation 3-6

All these basic functions are used to describe the statistics of the random field.

3.4 Random Fractal Theory

3.4.1 Fractional Brownian Motion

In one dimension fBm, $B_H(t)$, is a single valued function of time, t . Its increments $B_H(t_2) - B_H(t_1)$ have a Gaussian distribution:

$$f(x) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left[-\frac{(x-\mu)^2}{2\sigma^2}\right]$$

Equation 3-7

And variance:

$$\gamma(t) \propto |t_2 - t_1|^{2H}$$

Equation 3-8

The parameter H , known as the Hurst Exponent, has values between $0 \rightarrow 1$. A value of $H = 1/2$ is known as classical Brownian motion. The derivative of classical Brownian motion corresponds to uncorrelated Gaussian white noise and has independent increments. For $H > 1/2$ there is a positive correlation both for the increments and its derivative. For $H < 1/2$ there is a negative correlation. H is related to the fractal dimension D , by $H = 2 - D$. This relation states that the fractal dimension, D , lies somewhere between 1, a line, and 2, a plane. Since the variance depends only on the difference between t_2 and t_1 , and not the actual values, the increments are said to be stationary. The random field is also isotropic since all points and

directions are statistically equivalent. The random field also possesses a statistical scaling behavior known as self-affinity. That is, if the time scale t is changed by a factor, S , then the increments of the variance change by a factor, S^{2H} :

$$\gamma(St) = S^{2H}[\gamma(t)]$$

Equation 3-9

Hence, unlike statistically self-similar curves, fBm requires different scaling factors in the two coordinates.

3.4.2 Spectral Density

A random function, $R(t)$ in time is often characterized by the spectral density, $S(v)$. The spectral density gives frequency, v , information about the time correlation of $R(t)$. When $S(v)$ increases steeply at low v , $R(t)$ varies more slowly. It can be shown (Peitgen, 1988) that the power spectral density, $S(v)$, of fBm in one dimension has the following relation:

$$S(v) \propto \frac{1}{v^\beta} \propto \frac{1}{v^{5-2D}} \quad , \quad 1 < D < 2$$

Equation 3-10

Where $\beta = 2H + 1$ and fractal dimension, $D = 1 + (3 - \beta) / 2 = 2 - H$, $v = \sqrt{I_x^2}$, $I = 0, 1, 2$, ...size of X dimension.

This result agrees with the concepts of spectral density and Wiener-Khintchine relation to non-stationary random noises. $S(v)$ is non-zero for all frequencies indicating detail at all scales. As β

is decreased (higher D), higher values of S(v) at high v values occur and low values of S(v) occur at low v values. This results in a curve that is closer to a plane. As β is increased (lower D), lower values of S(v) at high v values occur and high values of S(v) occur at low v values. This results in a curve that is closer to a line.

Extending fBm into 2 and 3 dimensions has the following relations:

$$S(v) \propto \frac{1}{v^{\beta+1}} \propto \frac{1}{v^{8-2D}} \quad , 2 < D < 3$$

Equation 3-11

Where $v = \sqrt{I_x^2 + I_y^2}$, $I = 0, 1, 2 \dots$ size of dimension X and Y

$$S(v) \propto \frac{1}{v^{\beta+2}} \propto \frac{1}{v^{11-2D}} \quad , 3 < D < 4$$

Equation 3-12

Where $v = \sqrt{I_x^2 + I_y^2 + I_z^2}$, $I = 0, 1, 2 \dots$ size of dimension X, Y, and Z

In general for Euclidean dimension, d, $\beta = 2d - 2D + 3$ and $D = d + 1 - H$. I.e., d = **1**-line, **2**-surface, **3**-cloud, **4**-cloud time series, etc...and $0 < H < 1$ the Power Spectral Density, S(v) is:

$$S(v) \propto \frac{1}{v^{\beta+d-1}} \propto \frac{1}{\sqrt{(I_x^2 + I_y^2 + \dots I_d^2)^{2d-2D+3+(d-1)}}} \quad , d < D < d + 1$$

Equation 3-13

Where $v = \sqrt{I_x^2 + I_y^2 + \dots I_d^2}$, $I = 0, 1, 2 \dots$ size of dimension and d = Euclidean dimensionality

3.5 Fractional Brownian Motion Algorithm

There are three common algorithms for creating fBm: midpoint displacement, spectral synthesis, and turning bands. This section covers only spectral synthesis. For additional methods of creating fBm see Yin (1996). For a discussion of additional models to add small-scale turbulent detail see Stam (1991,1995).

3.5.1 Spectral Synthesis

fBm texture maps are used in this research to simulate turbulent motion on the microscale level. It is this turbulent motion that gives a gas cloud its pseudo-random shape. The following section discusses the computer algorithm used to create these texture maps. Some examples are included.

Spectral synthesis or the Fourier filtering method is based on the spectral density property of fBm. It uses the Fourier transform to create a process that has a spectral density as stated in Equation 3-13. The disadvantages of this process are possible large memory requirements, the whole spectrum must be created at one time, redundancy due to Fourier transform symmetries, and as addressed by Falconer (1990) the fBm approximation is poor when the frequency is very small. With this in mind the following is a diagram of pseudo code to create 3D fBm texture maps using the spectral synthesis technique:

1. Create a 3D Hermetian volume of random phases with values: $0 \rightarrow 2\pi$ (white noise).
Hermetian means a complex-valued function with the real part even and the imaginary part odd. $R(x) = R(-x)$, $I(x) = -I(-x)$

2. Create a 3D volume (same size as the volume of random phases) of amplitudes

proportional to $\frac{1}{(\sqrt{I_x^2 + I_y^2 + I_z^2})^{11-2D}}$. Where I = the integer index (0, 1, 2, ..., N-1)

of dimension x, y, z, N is the dimension size, and D is the fractal dimension with range, $3 < D < 4$. $11-2D$ represents the fractional Brownian exponent.

3. For efficiency use the inverse Fourier transform (IFT) with dimension N equal to a power of two and calculate the IFT(amplitudes*phases) = fBm, real valued and random due to the Hermetian symmetry properties.

The following figures illustrate the application of a fBm texture map to the Gaussian distribution. The 2D representation is a slice through the 3D distribution along the Z-axis. The 3D fBm has a fractal dimension of $D = 4$, variance = 40. The distributions have been scaled for viewing purposes.

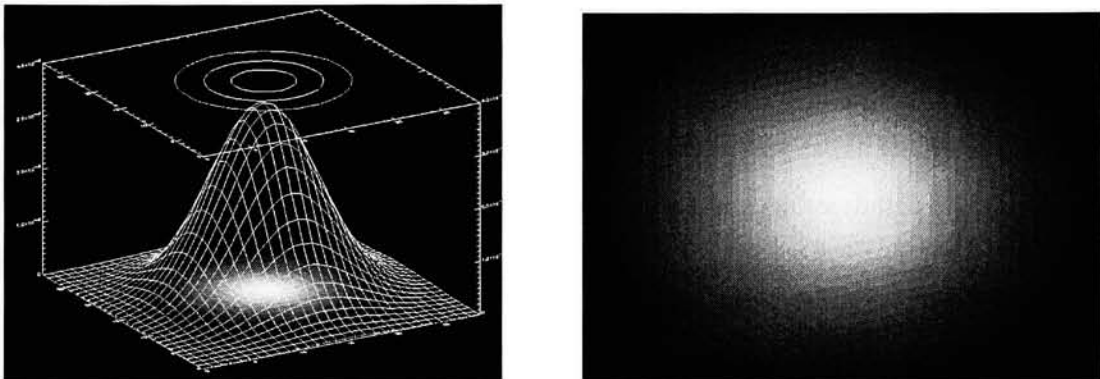


Figure 3-2 Gaussian 2D & 3D distributions, without texture map

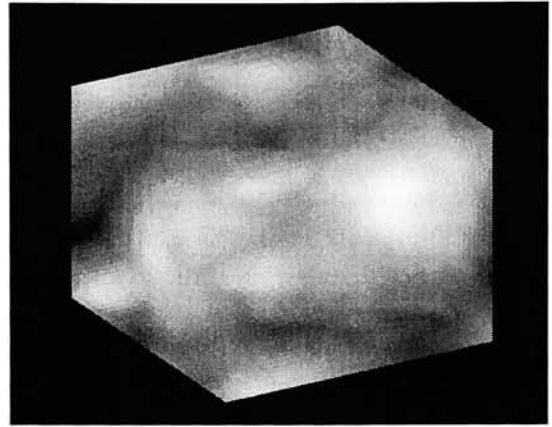
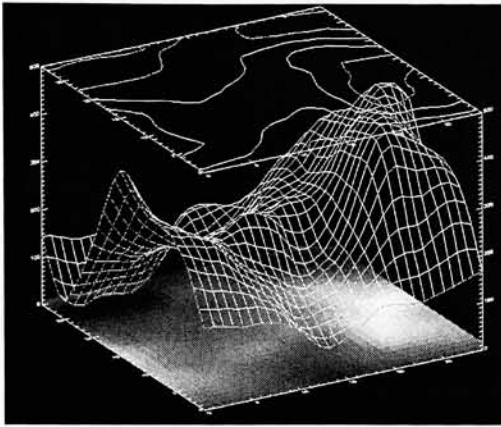


Figure 3-3 Fractional Brownian motion 2D & 3D texture maps

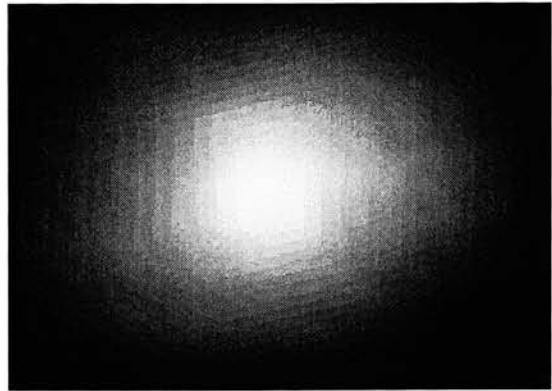
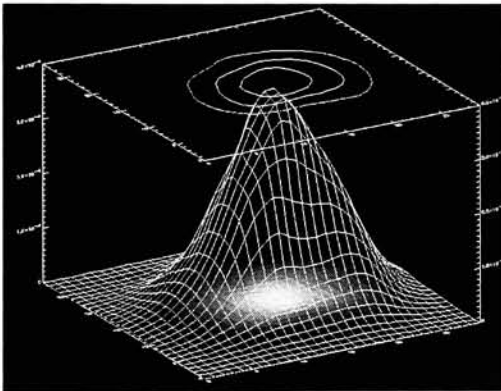


Figure 3-4 Gaussian*Fbm 2D & 3D distributions

Section 3 discussed the probability and implementation behind the creation of 3D fractional Brownian motion maps. These maps are used to create micro-scale variances, which simulate the turbulent nature of gas clouds.

4 Turbulent Diffusion

The following is a discussion of the principles behind turbulent diffusion or dispersion as it relates to smoke plumes. The principles used to model turbulent diffusion from discrete sources are assumed to be similar to processes affecting the evolution of a gas cloud. Therefore, these principles are the basis of the diffusion and advection model governing the gas cloud at time, $t > 0$. The following sections are taken from Blackadar (1997), Beychok (1994), and Briggs (1969) except where noted.

4.1 Conservation of Matter

A particle leaves a source at some time, t , and moves in response to turbulent airflow, with wind direction x , horizontal direction y , and vertical direction z . The probability of the particle lying between x and $x+dx$, y and $y+dy$ and z and $z+dz$ given independent probabilities and the particle does not change form is:

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} P(x)P(y)P(z)dx dy dz = 1$$

Equation 4-1

Where $P(x)$ is the probability density function of the particle lying between x and $x+dx$

An instantaneous point source of strength Q (mass in grams) has a concentration X (mass per unit volume). The expected amount of effluent in a volume of dimensions dx, dy, dz located at x, y, z is $Xdx dy dz$. The above statements give the following Equations:

$$X(x,y,z) = QP(x)P(y)P(z) \text{ [mass / volume]}$$

Equation 4-2

$$Q = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} X(x,y,z) dx dy dz \text{ [mass]}$$

Equation 4-3

The previous equations assume independence of probability distributions and conservation of matter. Wind changes with height, dry deposition, wet deposition, and chemical transformations undermine these assumptions. If the gas cloud is confined between two plates, i.e. the ground and an inversion layer above, then after a long time the concentration will approach a limit in which the concentration is uniformly distributed as a function of height between the plates, and zero above and below the plates.

Most instantaneous sources can be treated as single points. While the source is finite in size, they are relatively tiny in comparison to the size of the initial gas cloud. The gas cloud can be dealt with by assuming it represents the evolution from a virtual point source at an earlier time. Superimposing a sequence of instantaneous point sources spaced at time interval, dt , can create a continuous point source. [Blackadar, 1997] Refer to Figure 4-1.

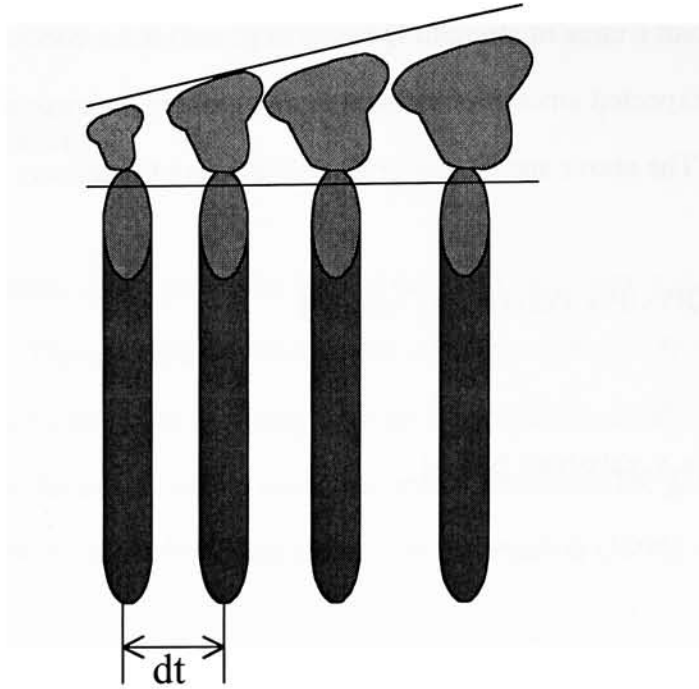


Figure 4-1 Instantaneous point sources spaced at time interval, dt , used to create a continuous source

The source strength, Q , must be redefined as an emission rate, Qdt . To simplify things it is assumed that no dispersion in the mean wind direction occurs. Thus the emission of each instantaneous source is confined within a slab of width $u dt$, where u is the wind speed (m/s), at a position, x , equal to u times the time since emission. Using the uniform probability between two plates case, the $P(x)$ for the concentration within the plates is:

$$P(x) = \frac{1}{u dt} [1 / m]$$

Equation 4-4

Where \bar{u} = mean wind speed [m/s]

Substituting into Equation 4-2 to get total concentration gives:

$$X(x,y,z) = \frac{Q}{u} P(y)P(z) \text{ [mass/volume]}$$

Equation 4-5

Hence concentration is proportional to emission rate and inversely proportional to wind speed.

4.2 Probability Density Functions: P(x), P(y), P(z)

The form of the PDFs has been left out of the previous section. It is important to know the PDFs form as a function of time and space. It is the PDF that describes the mean concentration of the gas cloud at any time and location. The earliest approach to a solution to this problem is given by K-theory or first-order closure credited to Schmidt (1925) and Prandtl (1925). Assuming the mean wind is constant in the x-direction and given the exchange coefficients K_x , K_y , K_z , the solution produces the differential Fick Equation:

$$\frac{\partial X}{\partial t} = K_x \frac{\partial^2 X}{\partial x^2} + K_y \frac{\partial^2 X}{\partial y^2} + K_z \frac{\partial^2 X}{\partial z^2} \left[\frac{\text{mass}}{\text{vol} * \text{time}} \right]$$

Equation 4-6

Where $X=0$ at $t=0$, $x,y,z \neq 0$ and $X \rightarrow 0$ as $t \rightarrow \infty$ for all x,y,z and Equation 4-3 holds where Q is the total mass of the gas at time, $t=0$.

The solution to this differential Equation for an instantaneous point source and coordinate system that moves with the mean wind is:

$$X = \frac{Q}{(2\pi)^{\frac{3}{2}} \sigma_x \sigma_y \sigma_z} \exp\left(\frac{-x^2}{2\sigma_x^2}\right) \exp\left(\frac{-y^2}{2\sigma_y^2}\right) \exp\left(\frac{-z^2}{2\sigma_z^2}\right)$$

$$= QP(x)P(y)P(z) \left[\frac{\text{mass}}{\text{volume}} \right]$$

Equation 4-7

The concentration for a continuous point source can be determined using Equation 4-5.

The quantities given by sigma indicate the width of the gas cloud. How the values change with respect to time is an important aspect in determining the gas cloud evolution. The observed values show an increase of sigma proportional to time, t, raised to the n, t^n , where $0.75 < n < 1.0$. The Fick Equation predicts an n of 0.5.

K-theory falls short because it is based on the assumption that the diffusion is carried on by the energy containing eddies or the longest wavelength eddies in the spectrum. However, when the gas cloud is small the entire spectrum of eddies causes diffusion. Thus the gas cloud expands more rapidly at the beginning than when it gets larger (see Figure 4-4).

4.3 The Gaussian Model

The form of the PDFs that Fick predicts is known as the Gaussian or normal distribution as shown in Figure 4-2:

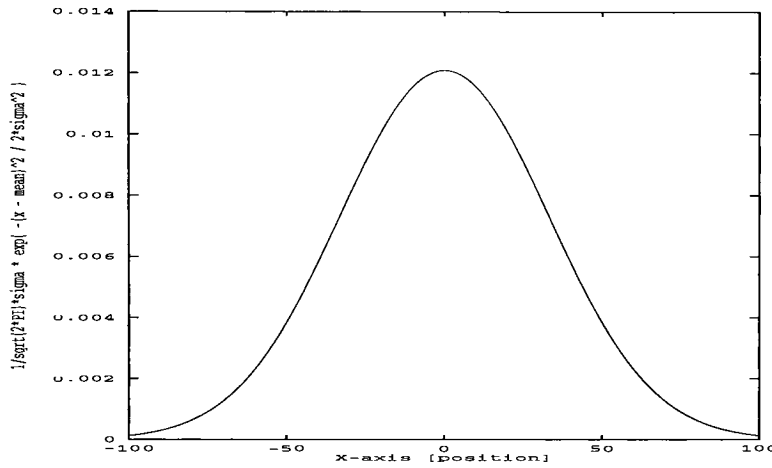


Figure 4-2 1D Gaussian distribution function

This shape for both gas plumes and clouds is consistent with most experimental concentration data if sufficient allowance is given to sample irregularities. It takes the form in one dimension:

$$P(y) = \frac{1}{(2\pi)^{1/2} \sigma_y} \exp\left(\frac{-(y-\mu)^2}{2\sigma_y^2}\right)$$

Equation 4-8

Where μ = mean, y = location in the y axis, and σ = the standard deviation

The 3D Gaussian is defined as the product of three 1D Gaussian functions:

$$\text{Gaus}(X, Y, Z) = \text{Gaus}(X) * \text{Gaus}(Y) * \text{Gaus}(Z)$$

Equation 4-9

Where $0 < X < N\sigma_x$, $0 < Y < N\sigma_y$, $0 < Z < N\sigma_z$, and N = number of stdev

As stated earlier, in the 1D case, a standard deviation of one sigma, σ_y , from the center line (mean interval) contains 66% of the PDF volume. A two-sigma deviation, $2\sigma_y$, contains 95% of the PDF volume. Clearly a way to accurately predict σ_y and σ_z as a function of wind speed, heating rate, and position is needed. The first approach to estimating σ_y and σ_z is to use Lagrangian statistics of particle motion. The second is to use experimental results based on field studies.

4.4 Dispersion Coefficients

4.4.1 Lagrangian Statistics of Particle Motion

Dispersion coefficients, σ_y and σ_z , were discussed by Taylor (1935) in which turbulence is treated as a continuous process, in contrast to K-theory which is based on a model of discrete mixing events. The turbulence is assumed stationary. Particles are released from a source one by one, and each particle's motion is independent of the one before it. See Figure 4-3.

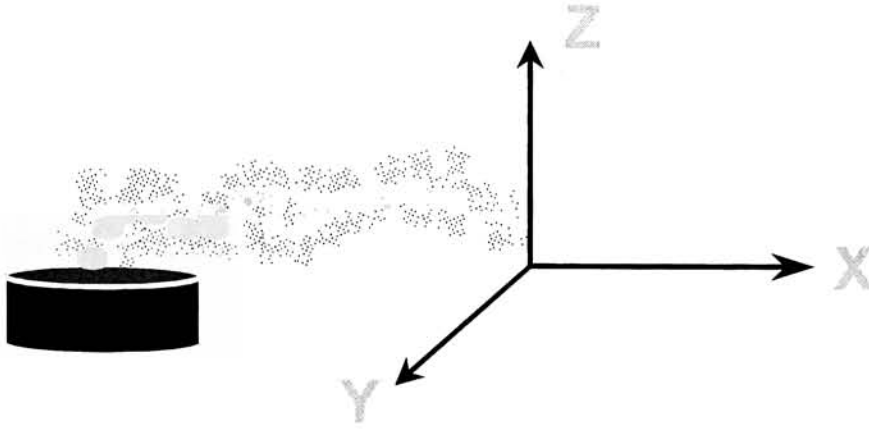


Figure 4-3 Particle motion from insantaneous source

Taylor predicted the variance of the position of all the particles at a time interval, T after leaving the source. It is based on the Lagrangian autocorrelation:

$$R(t_2) \equiv \frac{\overline{v(t)v(t+t_2)}}{\overline{v^2}}$$

Equation 4-10

Where $v(t) = dy/dt$ or dz/dt at time, t , and $v(t+t_2) =$ the value of the same particle at t_2 seconds later

The average is extended over all particles released from the same point. Under stationary conditions, the Lagrangian autocorrelation is a function of t_2 only. Its value for $t_2 = 0$ is one.

Integrating $R(t_2)$ from the moment of release ($t=0$) to a time t_2 seconds later results in an interpretation of Taylor's Equation as:

$$\sigma^2_{y,z}(X) \cong 2\sigma_v^2 \int_0^T dt \int_0^{t_2} R(t_2) dt_2$$

Equation 4-11

Where σ = dispersion coefficient, X is the mean distance from the point of release of all particles that have traveled a time T since leaving the source.

Thus the diffusion is reduced to finding R as a function of t_2 . This falls into two cases. The first is for small T such that $R(t_2) \approx 1$. This case gives:

$$\sigma_{y,z} \approx \sigma_v T \approx \frac{\sigma_v X}{\bar{\mu}} \approx \sigma_\alpha X$$

Equation 4-12

Where σ_v = standard deviation in dydt or dzdt, $\bar{\mu}$ is the mean wind speed, and σ_α = the standard deviation of wind directions measured by a wind vane situated at the point of release = $\sigma_v/\bar{\mu}$ for small angles

The second is for large T, where the Taylor Equation becomes:

$$\sigma_{y,z}^2 = 2\sigma_v^2 T_L T \quad \text{or} \quad \sigma_{y,z} \propto \sqrt{T}$$

Equation 4-13

Where T_L = the Lagrangian time scale:

$$T_L \equiv \lim_{t \rightarrow \infty} \int_0^t R(t_2) dt_2$$

Equation 4-14

This limiting behavior is identical to the Fick Equation. A spectral representation of these two cases is described by:

$$\sigma_{y,z}^2 = T^2 \int_0^{\infty} F_v(n) \frac{(\sin \pi n T)^2}{(\pi n T)^2} dn$$

Equation 4-15

Where $F_v(n)$ = Fourier transform of the autocorrelation and n = number of cycles per unit time experienced by a particle as it moves along its trajectory

The weighting function describes the Lagrangian energy spectrum responsible for spreading or diffusing the gas cloud. Figure 4-2 illustrates.

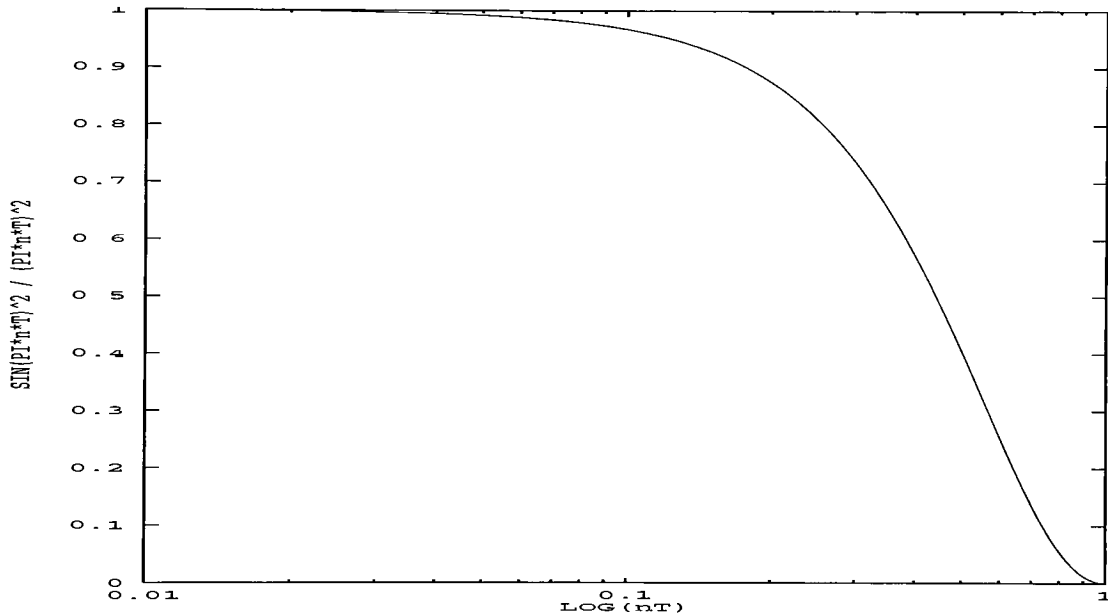


Figure 4-4 Taylor weighting function of energy spectrum that causes diffusion as a function of time T [Blackadar, 1997]

When T is very small which corresponds to the spreading of the plume just as it leaves the source the Taylor weighting function is one, and the integral is $\overline{v^2}$, the total variance of the y-components of the turbulent velocities. Thus, initially the entire energy spectrum participates in

the spreading of the plume. This Taylor weighting function is a filter that defines the portion of the Lagrangian spectrum that is responsible for the dispersion at time T after release. Figure 4-4 shows that only those frequencies for which the product, nT , is small are effective. Thus as T increases, the eddy sizes that are effective in increasing the width are more and more restricted to the longest periods. [Blackadar, 1997]

4.4.2 Pasquill Stability Classes and Experimental Dispersion Coefficients

From Lagrangian statistics it was shown that the single most important parameter for determining dispersion of the gas cloud was the standard deviation of the wind directions, σ_α . Estimating a meaningful parameter must consider the heating rate and wind speed. However, the heating rate is often unknown and so it is useful to use an experimentally derived classification scheme devised by Frank Pasquill.

The amount of turbulence in the air has a major effect upon on the rise and dispersion of a gas plume or cloud. This turbulence has been classified into defined increments or stability classes. The most widely used scheme is known as the Pasquill Stability Classes. It has six categories: A - very stable $\sigma_\alpha = 25$, B -unstable $\sigma_\alpha = 20$, C - slightly unstable $\sigma_\alpha = 15$, D - neutral $\sigma_\alpha = 10$, E - slightly stable $\sigma_\alpha = 5$, and F - stable $\sigma_\alpha = 2.5$. This scheme does not include very light wind speeds on a clear night. It also does not take into effect mixed-layer height or roughness. Increasing the roughness or decreasing the height increases the instability. Thus one way to compensate for this is to move to the next more unstable class. See Figure 4-5 for meteorological conditions that indicate class type.

| Pasquill Stability Class Related to Wind Speed and Insolation | | | | | |
|---|--------------------|----------|--------|-----------------------|------------------|
| Surface wind speed (m/s) | Daytime Insolation | | | Night-time Conditions | |
| | Strong | Moderate | Slight | $\geq 4/8$ cloud | $\leq 3/8$ cloud |
| < 2 | A | A-B | B | | |
| 2-3 | A-B | B | C | E | F |
| 3-5 | B | B-C | C | D | E |
| 5-6 | C | C-D | D | D | D |
| >6 | C | D | D | D | D |

Figure 4-5 Pasquill Stability Types [Beychok, 1994]

Where insolation (incoming solar irradiance): strong $> 598 \text{ W/m}^2$, moderate = $301\text{-}598 \text{ W/m}^2$, slight $< 301 \text{ W/m}^2$, and Neutral class D applies to heavily overcast skies, day or night.

According to Slade (1968) the Pasquill dispersion coefficients are reasonably accurate for assessing the magnitude of dispersion problems. However, the Pasquill dispersion coefficients are based on field experimental data that make them appropriate only for level terrain in open, rural areas. In an urban or industrial area, dynamic turbulence generated by buildings in the path of a dispersing plume or cloud tends to increase the spread. Buildings in the plume or cloud path also create "downwash" effects that tend to bring the plume to ground level more quickly than would be the case in an open area with unobstructed flow. Urban and industrial areas are also generally somewhat warmer than rural areas and, therefore, have more thermally generated turbulence that further affects plume dispersion. Therefore an Equation for urban dispersion coefficients is needed.

McMullen (1975) has published an analytical Equation for the experimental rural Pasquill coefficients:

$$\sigma = \exp[I + J * (\ln x) + K * (\ln x)^2]$$

Equation 4-16

Where σ = rural dispersion coefficient [m], x = downwind distance [km]

The constants I, J, and K are provided for both vertical, σ_z , and horizontal, σ_y , standard deviations in Figure 4-6.

| Pasquill Stability Class | σ_z | | | σ_y | | |
|--------------------------------|------------|--------|--------|------------|-------|--------|
| | I | J | K | I | J | K |
| A - very unstable | 6.035 | 2.1097 | .0770 | 5.357 | .8828 | -.0076 |
| B - unstable | 4.694 | 1.0629 | .0136 | 5.058 | .9024 | -.0096 |
| C - slightly unstable | 4.110 | .9201 | -.0020 | 4.651 | .9181 | -.0076 |
| D - neutral | 3.414 | .7371 | -.0316 | 4.230 | .9222 | -.0087 |
| E - slightly stable | 3.057 | .6794 | -.0450 | 3.922 | .9222 | -.0064 |
| F - stable | 2.621 | .6564 | -.054 | 3.933 | .9191 | -.0070 |

Figure 4-6 McMullen Equation Constants for Rural Diffusion [Beychok, 1994]

Experimental data has confirmed that urban areas have higher dispersion coefficients. This causes gas plumes and clouds to expand more rapidly and hence the maximum concentration occurs closer to the source.

Gifford (1975) developed an Equation for urban dispersion coefficients expressed as:

$$\sigma = (Lx)(1 + Mx)^N$$

Equation 4-17

Where σ = urban dispersion coefficient [m], x = downwind distance [km]

The constants L , M , and N are provided for both vertical, σ_z , and horizontal, σ_y , standard deviations in Figure 4-7.

| Pasquill Stability Class | σ_z | | | σ_y | | |
|--------------------------------|------------|-----|------|------------|------|------|
| | L | M | N | L | M | N |
| A - very unstable | 240 | 1.0 | .50 | 320 | 0.40 | .50 |
| B - unstable | 240 | 1.0 | .50 | 320 | 0.40 | -.50 |
| C - slightly unstable | 200 | 0.0 | 0.0 | 220 | 0.40 | -.50 |
| D - neutral | 140 | .30 | -.50 | 160 | 0.40 | .50 |
| E - slightly stable | 80 | 1.5 | .50 | 110 | 0.40 | -.50 |
| F - stable | 80 | 1.5 | .50 | 110 | 0.40 | -.50 |

Figure 4-7 Gifford Equation Constants for Urban Diffusion [Beychok, 1994]

The experimental equations assume that the dispersion in the wind direction is negligible. This is because the equations are assuming a continuous source. This research deals with discrete sources so the standard deviation in the x dimension is estimated as equivalent to the standard deviation in the y dimension. Since there is no experimental data for the spreading of a discrete source the statement is pure speculation and may or may not represent the dispersion of an actual cloud evolution. It appears to be reasonable for distances close to the source where the relationship is almost linear. However, at larger distances (>1.0 km) from the source using rural coefficients the dispersion in the y dimension compared to the dispersion in the z direction maybe to small. This same spreading at large distances occurs in the urban coefficients but to a far lesser degree.

4.4.3 Briggs Plume or Cloud Rise

Briggs developed Equations for the vertical rise component of 4 types of gas plumes:

1. Cold jet plumes in calm conditions
2. Cold jet plumes in windy conditions
3. Hot, buoyant plumes in calm conditions
4. Hot, buoyant plumes in windy conditions

A plume or cloud has vertical momentum due to an initial vertical velocity and its buoyancy if it is warmer than its surrounding ambient air. Buoyancy force is also known as Archimede's principle. A body that is partly or entirely submerged in a fluid (i.e. air) is buoyed up by a force equal in magnitude to the weight of the displaced fluid and directed upward along a line through the center of gravity of the displaced fluid.

The assumptions Briggs made were to neglect atmospheric friction and heat loss. Thus the plume or cloud conserves its original velocity and buoyancy momentum. A plume or cloud will grow as it entrains the ambient air. The entrainment of air is caused by self-induced turbulence from the plume or cloud's initial vertical velocity and buoyancy momentum. The plume or cloud will grow in mass but its momentum will be conserved since the total velocity will decrease. At some time, t_2 , later the vertical velocity component will reduce to zero and the plume or cloud will level out. In general the plume or cloud's rise is dependent on 5 conditions:

1. Initial vertical velocity
2. Initial buoyancy momentum
3. Wind velocity

4. Downwind distance from source
5. Atmospheric Pasquill stability class

For this research initial vertical velocity and momentum were estimated parameters since no initial blast kinematics were modeled. It was assumed that these parameters would be relatively small since the gas clouds in question are not as hot as factory gas plumes and are not rising from up a chimney.

The trajectory of the plume or cloud is broken up into three stages: initial, transition, and final. The initial stage is where air entrainment is due to self-induced turbulence. The final stage is where the air entrainment is due to atmospheric turbulence and is defined at the point, x^* . The maximum vertical height is determined at the distance $3.5 x^*$.

The next sections are a discussion of the Briggs' Equations for bent-over, hot buoyant plumes. The assumption is that stack exit velocities are in the range of 6-30 m/sec with exit temperatures in the range of 120-260 degrees Celsius. This may not be an appropriate assumption for nerve agent gas clouds (lower temperatures and velocities). However, it gives a starting point for the vertical rise of a gas cloud.

4.4.4 Briggs Buoyancy Flux

Briggs uses a parameter called buoyancy flux, F , to categorize exit buoyancy of the plume or cloud defined as:

$$F = \frac{F_B}{\pi \rho_a} [\text{m}^4/\text{s}^3]$$

Equation 4-18

Where F_B = buoyancy force [$\text{g}\cdot\text{m}/\text{s}^2$] and ρ_a = density of ambient air [g/m^3]

The buoyancy force is based on Archimede's principle. It is defined as:

$$F_B = gV_s(\rho_a - \rho_s) [\text{g}\cdot\text{m}/\text{s}^2]$$

Equation 4-19

Where g = gravitational constant [$9.807\text{m}/\text{s}^2$], V_s = stack gas flow [m^3/s] and ρ_s = density of stack gas

Assuming that the plume density is close enough to the ambient air density (known as the Boussinesq approximation) gives:

$$\rho_a T_a = \rho_s T_s$$

Equation 4-20

Where T_a = ambient air temperature [K] and T_s = temperature of gas [K]

The Boussinesq approximation allows one to neglect the effects of density fluctuations. In natural processes density fluctuations are dominated by temperature fluctuations. This assumption is questionable if the initial density difference between the plume or cloud and ambient air is large. Incorporation of added mass helps to explain differences in entrainment parameters determined from plume growth.

Combining Equation 4-18, Equation 4-19, Equation 4-20 yields the Briggs' buoyancy flux parameter F :

$$F = \left(\frac{gV_s}{\pi} \right) \frac{(T_s - T_a)}{T_s}$$

$$= (gv_s r^2) \frac{(T_s - T_a)}{T_s}$$

Equation 4-21

Where v_s = stack exit velocity [m/d] and r^2 = stack exit area [m²]

4.4.5 Briggs Stability Parameter

Briggs defines a stability parameter, s , also known as the restoring force that indicates the effect atmospheric turbulence has on the plume or cloud rise. It is defined as:

$$S = \left(\frac{g}{T_a} \right) \frac{d\theta}{dz} [1/s^2]$$

Equation 4-22

Where $d\theta/dz$ = potential temperature gradient defined by:

$$\frac{d\theta}{dz} = \text{ambient temperature gradient} \quad \text{dry adiabatic lapse rate}$$

$$= \frac{dT}{dz} - \Gamma$$

Equation 4-23

Where $\Gamma = -0.01002$ [K/m] and dT/dz is defined for each Pasquill stability class as shown in Figure 4-8:

| Pasquill Stability Class | Ambient Temperature Gradient (dT/dz) [K/m] | Potential Temperature Gradient (dθ/dz) [K/m] |
|--------------------------|--|--|
| A - very unstable | < -0.01896 | -0.00893 |
| B - unstable | -0.01896 to -0.01695 | -0.00693 |
| C - slightly unstable | -0.01695 to -0.01495 | -0.00492 |
| D - neutral | -0.01495 to -0.00492 | 0.00510 |
| E - slightly stable | -0.00492 to 0.01495 | 0.02497 |
| F - stable | > 0.01495 | 0.02497 |

Figure 4-8 Ambient Temperature Gradient

Original tables were in °F/ft, unit conversion used to convert from °F/ft to K/m:

$$[1 \text{ } ^\circ\text{F} / \text{ft}] * [1\text{ft} / .3048 \text{ m}] * [1 \text{ K} / 1.8 \text{ } ^\circ\text{F}] = [\text{K/m}]$$

4.4.6 Briggs Plume Rise Δh and Δh_{\max} for Hot, Buoyant Bent-over Plumes

The following Equations are the Briggs plume rise 1972 Equations categorized by Pasquill stability class. They are for use with bent-over, hot buoyant plumes which consists of typical fossil fuel combustion products with an average molecular weight and specific heat that are the same as the ambient air. Plume rise is defined as the rise of the plume centerline or center of

mass above the origin. Buoyant plume describes a plume in which the effect of the initial velocity momentum is small. A bent over plume is one in which the vertical velocity of the plume is much smaller than the wind (horizontal) velocity.

The basic theory makes the following assumptions:

1. Buoyancy is conserved. I.e., motion is considered adiabatic - no heat loss:

$$\frac{d\theta_p}{dt} = 0$$

Equation 4-24

Where θ_p = local potential temperature in plume at h = height [m], t = time [s]

2. Pressure forces are small and have little effect on plume motion:

$$\frac{d\vec{v}_p}{dt} = \frac{g}{T} \theta' \vec{k}$$

Equation 4-25

Where \vec{v}_p = local velocity of gas in plume [m/s], g = gravitational constant [9.8 m/s²], T = temperature ambient air [K], $\theta' = \theta_p - \theta$ - temperature of air at height, h [K], \vec{k} = unit vector in vertical direction

3. Molecular viscosity is negligible due to plume Reynolds number being high thus local density changes are neglected:

$$\nabla \cdot \rho_p \vec{v}_p = 0$$

Equation 4-26

$$\text{Where } \nabla = \text{gradient operator} = x \frac{\delta}{\delta x} + y \frac{\delta}{\delta y} + z \frac{\delta}{\delta z}, \rho_p = \text{local gas density [g/m}^3\text{]}$$

Again, it is pointed out that gas clouds consisting of nerve agents may or may not follow these parameters. However, this is a starting point for the physics involved in gas cloud evolution due to turbulence, buoyancy, advection, and momentum. Section 4.4.7 will discuss what Briggs calls cold, jet plumes. He defines cold jet plumes as non-buoyant plumes whose plume rise is dominated by their initial velocity momentum.

The following Equations reference buoyancy flux, F (Equation 4-21), and stability parameter, s (Equation 4-22):

For Pasquill Stability Classes: A - very unstable, B - unstable, C - slightly unstable, and D neutral

For $F \geq 55 \text{ [m}^4/\text{s}^3\text{]}$:

$$\Delta h = \frac{1.6F^{\frac{1}{3}}x^{\frac{2}{3}}}{u} \text{ [m]} \quad \text{for } x < x_f$$

$$\Delta h_{\max} = \frac{1.6F^{\frac{1}{3}}x_f^{\frac{2}{3}}}{u} = \frac{38.7F^{\frac{3}{5}}}{u} \text{ [m]} \quad \text{for } x \geq x_f$$

Equation 4-27

For $F < 55 \text{ [m}^4/\text{s}^3\text{]}$:

$$\Delta h = \frac{1.6F^{\frac{1}{3}}x^{\frac{2}{3}}}{u} \text{ [m]} \quad \text{for } x < x_f$$

$$\Delta h_{\max} = \frac{1.6F^{\frac{1}{3}}x_f^{\frac{2}{3}}}{u} = \frac{21.4F^{\frac{3}{4}}}{u} \text{ [m]} \quad \text{for } x \geq x_f$$

Equation 4-28

For Pasquill Stability Classes: E - slightly stable and F - stable

For $1.8 u \text{ s}^{-1/2} \geq x_f$:

$$\Delta h = \frac{1.6F^{\frac{1}{3}}x^{\frac{2}{3}}}{u} \text{ [m]} \quad \text{for } x < x_f$$

$$\Delta h_{\max} = \frac{1.6F^{\frac{1}{3}}x_f^{\frac{2}{3}}}{u} = \frac{38.7F^{\frac{3}{5}}}{u} \text{ [m]} \quad \text{for } x \geq x_f \text{ and } F \geq 55 \text{ [m}^4/\text{s}^3\text{]}$$

$$\Delta h_{\max} = \frac{1.6F^{\frac{1}{3}}x_f^{\frac{2}{3}}}{u} = \frac{21.4F^{\frac{3}{4}}}{u} \text{ [m]} \quad \text{for } x \geq x_f \text{ and } F < 55 \text{ [m}^4/\text{s}^3\text{]}$$

Equation 4-29

For $1.8 u s^{-1/2} < x_f$:

$$\Delta h = \frac{1.6 F^3 x^{\frac{1}{2}}}{u} \text{ [m]} \quad \text{for } x < \frac{1.84u}{\sqrt{s}}$$

$$\Delta h_{\max} = 2.4 \left(\frac{F}{us} \right)^{\frac{1}{3}} \text{ [m]} \quad \text{for } x \geq \frac{1.84u}{\sqrt{s}}$$

Equation 4-30

Where Δh = initial plume rise [m], Δh_{\max} = max plume rise [m], x = downwind distance from source [m], x^* = distance to final stage [m], x_f = distance to max plume rise [m] = $3.5x^*$, $3.5x^* = 119F^{4/5}$ for $F \geq 55 \text{ [m}^4/\text{s}^3]$, $3.5x^* = 49F^{5/8}$ for $F < 55 \text{ [m}^4/\text{s}^3]$, u = wind velocity [m/s], F = buoyancy parameter $\text{[m}^4/\text{s}^3]$, and s = stability parameter $[1/\text{s}^2]$

4.4.7 Briggs Plume Rise Δh and Δh_{\max} for Cold, Jet Bent-over Plumes

Briggs defines plume rise for cold, bent-over jet plumes according to Pasquill stability class using the same principles discussed under hot, bent-over buoyant plumes. The major difference being jet plumes are dominated by initial velocity momentum. Unheated plumes composed mostly of air fit into this category.

Briggs defines a momentum flux parameter to account for the velocity momentum (mass*velocity):

$$\begin{aligned}
F_m &= \frac{\text{momentum}}{\pi \rho_a} = \frac{(\rho_s V_s) v_s}{\pi \rho_a} = \frac{(\rho_s \pi r^2 v_s) v_s}{\pi \rho_a} \\
&= \frac{(\rho_s r^2 v_s^2)}{\rho_a} \text{ Using the Boussinesq Approximation becomes} \\
&= \left(\frac{T_a}{T_s} \right) r^2 v_s^2 \text{ [m}^4/\text{s}^2]
\end{aligned}$$

Equation 4-31

Where T_a = ambient air temperature [K], T_s = plume gas temperature [K], r = stack exit radius [m], v_s = stack exit velocity [m/s], V_s = stack gas flow [m³/s], ρ_s = stack gas density [g/m³], ρ_a = ambient air density [g/m³]

Briggs Equations for bent-over, cold jet plume rise with Pasquill stability classes A, B, C, and D (unstable to neutral) are defined as:

$$\begin{aligned}
\Delta h &= \frac{2.3(F_m x)^{\frac{1}{3}}}{u^{\frac{2}{3}}} \\
\Delta h_{\max} &= \frac{6rv_s}{u}
\end{aligned}$$

Equation 4-32

Where F_m = momentum flux parameter [m⁴/s²], x = downwind distance [m], u = wind speed [m/s], r = stack radius [m], v_s = stack exit velocity [m/s]

Briggs Equations for bent-over, cold jet plume rise with Pasquill stability classes E and F (stable to very stable) are defined as:

$$\Delta h_{\max} = \frac{1.5 F_m^{\frac{1}{3}}}{u^{\frac{1}{3}} s^{\frac{1}{6}}} \text{ [m]}$$

Equation 4-33

Where Briggs has not defined Δh in the literature for this type of plume.

4.4.8 Briggs Plume Rise Δh and Δh_{\max} for Vertical Plumes

Vertical plumes arise when the horizontal wind velocity is negligible. Briggs (1969) defines the maximum plume rise for Pasquill class E and F, vertical, hot buoyant plumes as:

$$\Delta h_{\max} = \frac{5.0 F_m^{\frac{1}{4}}}{s^{\frac{3}{8}}} \text{ [m]}$$

Equation 4-34

Briggs (1969) defines the maximum plume rise for Pasquill class E and F, vertical, cold jet plumes as:

$$\Delta h_{\max} = \frac{4.0 F_m^{\frac{1}{4}}}{s^{\frac{1}{4}}} \text{ [m]}$$

Equation 4-35

4.5 Determining Cloud Dispersion and Rise Parameters

Sections 4.1 – 4.4 established the theory on which the Gaussian model is based on.

Make_blob.cc is a program written in C++ based on the experimental data and equations presented by Pasquill and Briggs in the preceding sections. Pasquill discusses how to determine the standard deviation to be used in the Gaussian equation. Briggs examines how to determine the rise associated with a cloud based on initial momentum and temperature. Both the standard deviation and rise are associated with Pasquill classes. These Pasquill classes incorporate the effect of turbulence on the cloud. Make_blob.cc reads in a data file that describes the environmental conditions (i.e., wind speed, insolation, rate of release, cloud cover, starting x, y, z coordinates, see Appendix A for a complete listing of input parameters). It then uses this information to determine the Pasquill class, standard deviation of the Gaussian PDF, and the rise (based on the Brigg's bent, over hot plume equations). Make_blob.cc then outputs a Gaussian PDF and final x, y, z coordinates according to the Pasquill and Brigg's equations. For completeness, cold jet plumes and hot, buoyant plumes in calm conditions are discussed but are not included in make_blob.cc.

5 Limitations of Gaussian Models

This section discusses the assumptions and constraints involved with using the Gaussian dispersion equations for modeling gaseous clouds from a single point source in flat terrain.

5.1 Assumptions and Constraints

The nature of modeling gas clouds involves certain assumptions about the conditions that effect the evolution of the cloud. These constraints are used to limit the number of variables involved and hence simplify the complexity of the problem. The drawback to these assumptions is that the prediction by the simulation will differ from the real cloud evolution. The addition of more variables may produce a better prediction but at the cost of computational time. Even if every variable could be included there would still be an inherent randomness or chaotic nature that the simulation would not be able to predict.

The Gaussian dispersion equations assume the following:

1. The wind speed and direction are constant throughout the evolution.
2. Atmospheric turbulence does not change throughout the evolution.
3. The mass of the plume is conserved.
4. Only crosswind and vertical dispersion occurs. This research assumes the downwind dispersion is the same as the crosswind dispersion.

5. The cloud PDF expands with a Gaussian distribution. The evolution of a cloud could have other distributions depending on the initial blast conditions, wind changes, turbulence factors, etc...

6. Using either urban dispersion coefficients or rural dispersion coefficients accommodates for terrain conditions. The basic Gaussian dispersion equation was not meant to handle coastal terrain, valleys, mountains, etc...

In short, the Gaussian models assume an ideal steady state of constant meteorological conditions over long distances, idealized plume geometry, uniform flat terrain, complete conservation of mass, and exact Gaussian distribution. Such ideal conditions rarely occur. [Beychok]

It is important to mention that these assumptions make the basic Gaussian model a simple tool that can assist the user but does not provide exact answers. The model is a start but can be improved upon. Some of these improvements are discussed in the section titled: Conclusion and Improvements.

6 Rendering

6.1 Raytracing

For a more complete explanation on raytracing the reader can refer to almost any text on computer graphics and rendering.

Raytracing uses the principles of geometric optics to traverse the path propagated by light beams (rays) from the sensor to the source. The sensor will have some plane of view to reduce the number of rays that need to be cast or sent out into the scene. The plane of view can be broken up into a rectangular grid. See Figure 6-1.

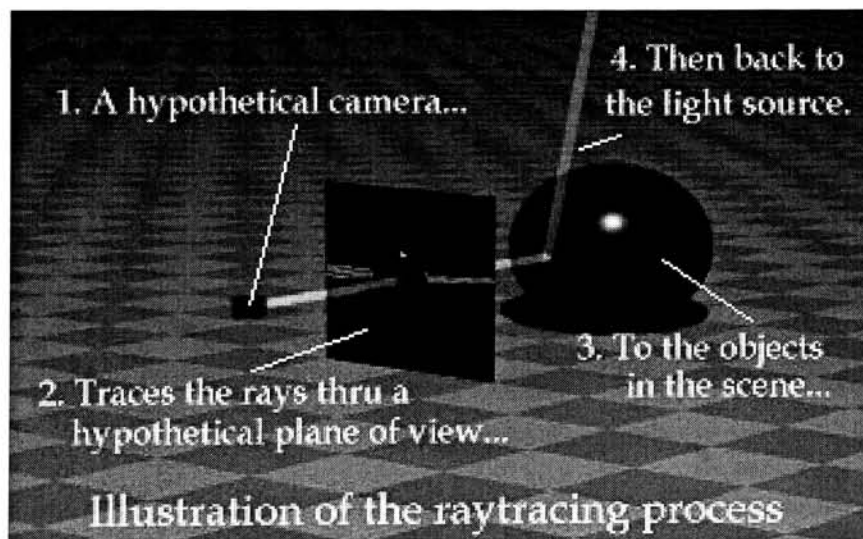


Figure 6-1 Simple Raytracing Process [Fusner, 1999]

Any two points in space define a vector. That vector must go through the plane of view in order to be used. The vector is then extended out to the scene, i.e. an AutoCAD drawing, to determine

if it intersects an object. If no intersection occurs then a default radiance is assigned. When an intersection occurs then a ray is sent to the illumination sources to determine the radiance falling on that point. Other optical properties that may be calculated include single or multiple reflections, transparency, refraction, diffusion, multiple sources, scattering, spectral responsivity of sensor, and antialiasing. Most of these optical properties include additional rays to be cast and hence longer computing times. It is not uncommon for raytracing runs to take many hours to several days depending on the complexity of the scene. This research used the raytracing package DIRSIG to calculate all radiance values using the “Big equation” as determined by the sensor / target geometry. Additional information on how to implement DIRSIG can be obtained from reading Brown (1999).

6.2 3D-DDA

A field of discrete regularly spaced items known as voxels represents the modeled gas cloud. A method is needed to determine “concentrations” along a ray through the voxelized gas cloud. The concentration can be determined along a ray by finding intersections within the field of regularly spaced voxels and summing the discrete values found within those intersecting voxels. There are various methods used. These methods determine which pixels a ray will visit when cast into the scene. One such method is called the three dimensional digital differential analyzer (3D-DDA), Fujumoto (1986). Another method is Cohen, "Voxel traversal along a 3D line", Heckbert (1994) which visits all voxels pierced by a 3D continuous line. The main difference between the two is Cohen's algorithm uses only integers and has no trigonometric function initialization, which reduces overall processing time. The following section discusses the 3D-DDA.

The purpose of the 3D-DDA is to represent an "optimal" line in three-dimensional space using discrete points to approximate a continuous line. The optimal line is one which is closest to the continuous, non-integer point line (for cases where the slope is not ± 1 , 0, or ∞) and satisfies the condition that only one pixel per column is chosen. The computer criteria for a line is outlined in Newman (1979):

1. Line should appear straight
2. Line should terminate accurately
3. Line should have constant density
4. Line density should be independent of line length and angle.
5. Line should be generated quickly.

The DDA is based on the differential Equation for a line:

$$\frac{dy}{dx} = \frac{\Delta y}{\Delta x}$$

Equation 6-1

It works by simultaneously incrementing x and y by small steps, C, proportional to the first derivative. In the continuous case this results in incrementing x and y by $C\Delta x$ and $C\Delta y$. The discrete 2D case works something like this:

To determine the next discrete pixel coordinate:

1. Determine driving axis (DA) by selecting the axis with the largest delta. I.e. $\Delta x > \Delta y$ then x is DA.
2. Unconditionally increment the DA by one pixel.

3. Update error term $\epsilon = \epsilon + \frac{\Delta y}{\Delta x}$, ϵ - measured perpendicular to DA, represents distance between optimal line and actual pixel generated. $+\epsilon$ = optimal line is above current point, increment passive axis (PA), set $\epsilon = \epsilon - 1$, $-\epsilon$ optimal line is below current point, don't increment PA.
4. Repeat steps 2 & 3.

To extend this 2D case to a 3D case two DDAs are linked to a single driving axis, see Figure 6-2. The DA now has two PAs. The DA is always incremented and the passive indices are generated for each plane. The passive indices of each plane share a common driving index. The 3D line is simply a projection of the two passive indices onto the driving plane that generates the x, y, and z coordinate. This method does not enumerate all pierced cells but only those that lie closest to the ideal line. However, Fujimoto (1986) does discuss how to enumerate all cells pierced by the line.

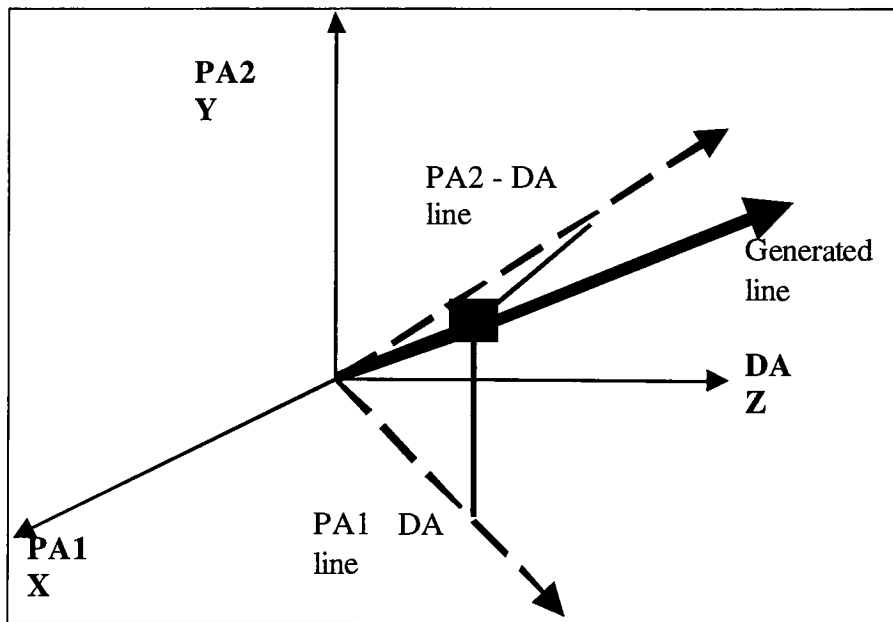


Figure 6-2 3D-DDA

6.2.1 Ray Tracing Gas Clouds in DIRSIG

A sensor pointed at a gas cloud will cast rays that intersect the gas cloud. The ray tracer must then determine how many regions the cloud is composed of. A gas cloud with several regions requires multiple calculations. Those calculations are dependent on the distance traveled, transmission of the cloud, emissivity of the cloud, and self-emission of the cloud. Again, scattered radiance is considered negligible compared to self-emission for $\lambda > 2.5 \mu\text{m}$ and is ignored in this research. Figure 6-3 illustrates a rectangular bounded gas cloud with three distinct regions.

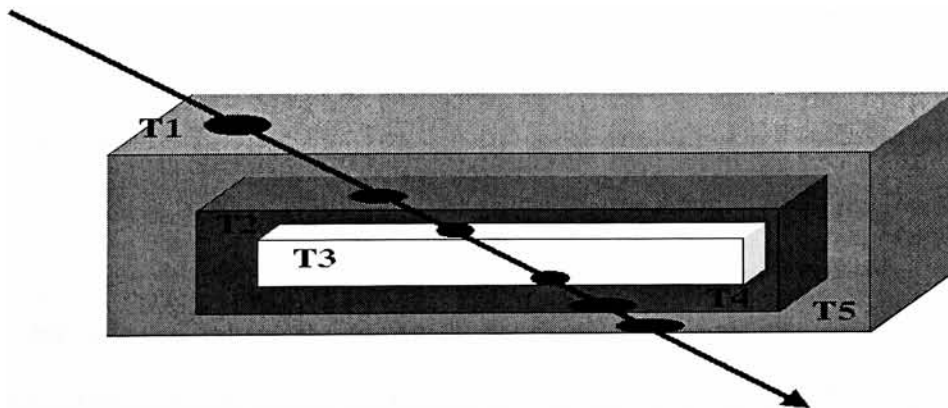


Figure 6-3 Ray tracing a gas cloud

These three regions are broken into 5 separate areas T1-T5. The total transmission through the gas cloud is:

$$T_{\lambda_{gas}} = T1_{\lambda} * T2_{\lambda} * T3_{\lambda} * T4_{\lambda} * T5_{\lambda}$$

Equation 6-2

Where T# = transmission through area T# as a function of wavelength

The emissivity through the gas cloud is:

$$\epsilon_{\lambda_{gas}} = 1 - T_{\lambda_{gas}}$$

Equation 6-3

The attenuation of radiance through the plume is given as:

$$L_{\lambda_{gas}} = L1_{\lambda} + L2_{\lambda} * T1_{\lambda} + L3_{\lambda} * T1_{\lambda} * T2_{\lambda} + L4_{\lambda} * T1_{\lambda} * T2_{\lambda} * T3_{\lambda} \\ + L5_{\lambda} * T1_{\lambda} * T2_{\lambda} * T3_{\lambda} * T4_{\lambda} \quad \left[\frac{W}{m^2 sr} \right]$$

Equation 6-4

Where L_{gas} = total radiance leaving plume, L# = radiance from each area T1-T5 as a function of wavelength

The cloud PDF (Gaussian distribution) defines the inverse volume used to calculate the concentration over the path length in question. The rigorous form for the cloud PDF over the length of the cast ray is given as:

$$\text{Cloud PDF}(i) \left[\frac{1}{m^3} \right] = \{ \text{for } i = 0, N \} \text{Gaus}(X, Y, Z)_i$$

Equation 6-5

Where N = length of cast ray in pixels, $\text{Gaus}(X,Y,Z)_i$ = value of Gaussian distribution at the X, Y, and Z coordinates associated with the ray at pixel i

This rigorous treatment would allow for a stepwise calculation of radiance:

$$L_{\text{sw}} = \sum_{i=0}^N [(1 - \tau_i) * L_{\text{BBi}} * \prod_{j=i+1}^N \tau_j]$$

Equation 6-6

Where N = length of cast ray in pixels, τ_i = transmission through voxel i, and L_{BBi} = black body radiance of voxel i

For this research the central limit theorem (CLT) was used to approximate the cloud PDF as given by:

$$\overline{\text{Cloud PDF}} \left[\frac{1}{\text{m}^3} \right] = \frac{1}{N} \sum_{i=0}^{N-1} \text{Cloud PDF}_i$$

Equation 6-7

Where $\overline{\text{Cloud PDF}}$ = the average Gaussian value over the length of the ray

The CLT column density [ppm-meters] or concentration over a given length through the gas cloud is calculated as:

$$\begin{aligned} \overline{\text{Column Density}} [\text{ppm} - \text{m}] = & \frac{\text{Mass} [\text{g}] * \overline{\text{Cloud PDF}} \left[\frac{1}{\text{m}^3} \right]}{\text{Molecular Weight of gas} \left[\frac{\text{g}}{\text{mol}} \right]} * \\ & \text{ideal Volume of gas adjusted for temp \& pressure} \left[\frac{\text{m}^3}{\text{mol}} \right] * \\ & \text{path_length} [\text{m}] * 1000000 \end{aligned}$$

Equation 6-8

The CLT gas cloud temperature is calculated as:

$$\overline{\text{Temp}} [^{\circ}\text{C}] = (\overline{\text{Cloud PDF}} * \text{gas exit temp} [^{\circ}\text{C}]) + ((1 - \overline{\text{Cloud PDF}}) * \text{ambient air temp} [^{\circ}\text{C}])$$

Equation 6-9

This produces a plume that is hotter in the center where the concentration of gas is higher and cools off to the ambient temperature at the edges where the concentration of gas is lower.

To summarize, the radiance through the gas cloud using the central limit theorem is defined as:

$$\overline{\text{Absorption}} = \overline{\text{concentration}} [\text{ppm}] * \overline{\text{absorption coefficients}} [1/\text{ppm} - \text{m}] * \overline{\text{pathlength}} [\text{m}]$$

$$\overline{\text{Transmission}} = e^{-\overline{\text{Absorption}}} \text{ or } 10^{-\overline{\text{Absorption}}}, \text{ see figure 6 - 4}$$

$$\overline{\text{Emmisivity}} = 1 - \overline{\text{Transmission}}$$

$$\overline{L}_{\text{cloud}} = \overline{L}_{\text{blackbody}} * \overline{\text{Emmisivity}}$$

Equation 6-10

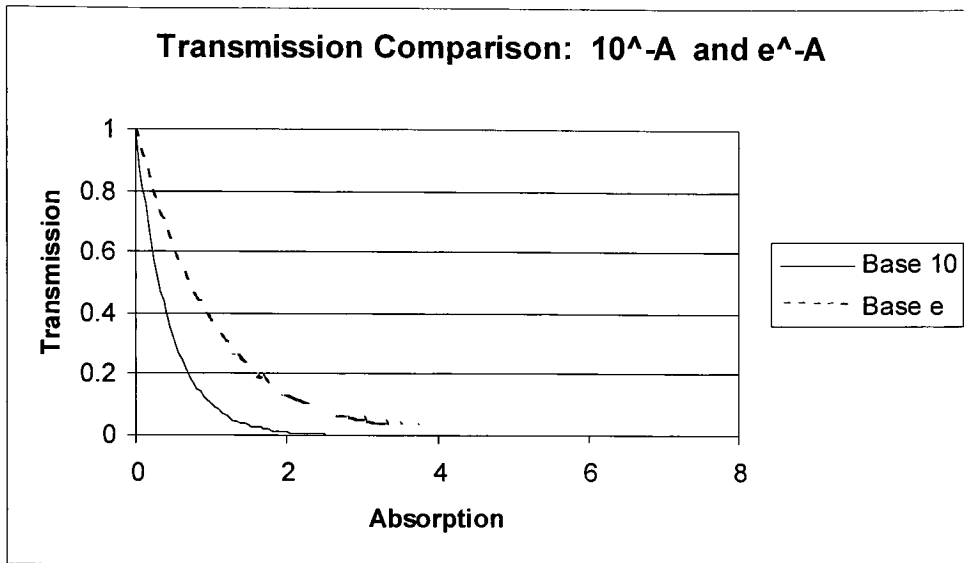


Figure 6-4 Transmission: base 10 vs base e

Transmission can be represented using base 10 or base e as shown in Figure 6-4. The decision on what form to use is dependent on how the data was acquired. If the data uses optical density then transmission is given in base 10. If the data uses optical depth then transmission is given by base e. This research used base 10 calculations.

For this research all cloud calculations are based on the central limit theorem. This could explain why the fBm texture and absorption / emission is not as pronounced as expected since the variations are averaged out. Essentially the cloud as shown in Figure 6-3 is treated as one region versus a stepwise treatment that would have many regions

If the cloud was broken into many regions such as a hot center surrounded by a cool shell the overall radiance would show more spectral and spatial structure. A stepwise version will be included in the future to address this problem.

The following 3 figures illustrates the differences between a CLT approximation and a stepwise treatment for calculating cloud radiance. The example uses a 1D case where the concentration is Gaussian and broken into 21 cells.

| | | | |
|-----------------------|-------------------------------|------------------------------|--------------------------------|
| Starting Temp: | | | |
| Gas Cloud [K] | Ambient [K] | Total path length [m] | Wavelength [um] |
| 350 | 300 | 21 | 10 |
| | Gaussian Concentration | Temp Values | Absorption Coef |
| Position | for Each Cell [ppm] | for Each Cell [K] | for Each Cell [1/ppm-m] |
| -10 | 0.000514093 | 300.0257046 | 1 |
| -9 | 0.001477283 | 300.0738641 | 1 |
| -8 | 0.003798662 | 300.1899331 | 1 |
| -7 | 0.00874063 | 300.4370315 | 1 |
| -6 | 0.017996989 | 300.8998494 | 1 |
| -5 | 0.033159046 | 301.6579523 | 1 |
| -4 | 0.054670025 | 302.7335012 | 1 |
| -3 | 0.080656908 | 304.0328454 | 1 |
| -2 | 0.106482669 | 305.3241334 | 1 |
| -1 | 0.125794409 | 306.2897205 | 1 |
| 0 | 0.13298076 | 306.649038 | 1 |
| 1 | 0.125794409 | 306.2897205 | 1 |
| 2 | 0.106482669 | 305.3241334 | 1 |
| 3 | 0.080656908 | 304.0328454 | 1 |
| 4 | 0.054670025 | 302.7335012 | 1 |
| 5 | 0.033159046 | 301.6579523 | 1 |
| 6 | 0.017996989 | 300.8998494 | 1 |
| 7 | 0.00874063 | 300.4370315 | 1 |
| 8 | 0.003798662 | 300.1899331 | 1 |
| 9 | 0.001477283 | 300.0738641 | 1 |
| 10 | 0.000514093 | 300.0257046 | 1 |
| Average | 0.047598199 | 302.37991 | |
| Sum | 0.999562187 | | |

Figure 6-5 CLT vs Step Wise calculations for cloud radiance, part 1 of 3

| Wavelength [um] | | | h | c | k | | |
|-----------------|--------------------|--------------------|--------------------|--------------------|------------------------------|--------------------|--------------------|
| 10 | | Base 10 | 6.63E-34 | 3.00E+08 | 1.38E-23 | | |
| Pathlength for | Absorption | Transmission | Emissivity | BB Radiance | Cloud Radiance | Transmission Cell | Radiance of Cell |
| Each Cell [m] | for Each Cell | for Each Cell | for Each Cell | for Each Cell | for Each Cell | to Edge of Cloud | to Edge of Cloud |
| 1 | 0.000514093 | 0.998816957 | 0.001183043 | 9.888844425 | 0.011698924 | 1 | 0.011698924 |
| 1 | 0.001477283 | 0.996604209 | 0.003395791 | 9.89652979 | 0.033606543 | 0.998816957 | 0.033566785 |
| 1 | 0.003798662 | 0.991291399 | 0.008708601 | 9.915066851 | 0.086346362 | 0.995425184 | 0.085951343 |
| 1 | 0.00874063 | 0.980075133 | 0.019924867 | 9.954599009 | 0.198344059 | 0.986756423 | 0.195717274 |
| 1 | 0.017996989 | 0.959407284 | 0.040592716 | 10.02889493 | 0.407100089 | 0.967095433 | 0.393704636 |
| 1 | 0.033159046 | 0.926490464 | 0.073509536 | 10.15130234 | 0.746217523 | 0.927838402 | 0.692369275 |
| 1 | 0.054670025 | 0.881718545 | 0.118281455 | 10.32647937 | 1.221431009 | 0.859633432 | 1.04998293 |
| 1 | 0.080656908 | 0.830506607 | 0.169493393 | 10.54047613 | 1.786541063 | 0.757954739 | 1.354117264 |
| 1 | 0.106482669 | 0.782559434 | 0.217440566 | 10.75571777 | 2.338729358 | 0.629486418 | 1.472198367 |
| 1 | 0.125794409 | 0.74852376 | 0.25147624 | 10.918346 | 2.745704593 | 0.492610535 | 1.35256301 |
| 1 | 0.13298076 | 0.736239713 | 0.263760287 | 10.97923033 | 2.895884941 | 0.36873069 | 1.067801654 |
| 1 | 0.125794409 | 0.74852376 | 0.25147624 | 10.918346 | 2.745704593 | 0.271474178 | 0.745387897 |
| 1 | 0.106482669 | 0.782559434 | 0.217440566 | 10.75571777 | 2.338729358 | 0.203204872 | 0.475241201 |
| 1 | 0.080656908 | 0.830506607 | 0.169493393 | 10.54047613 | 1.786541063 | 0.15901989 | 0.284095563 |
| 1 | 0.054670025 | 0.881718545 | 0.118281455 | 10.32647937 | 1.221431009 | 0.132067069 | 0.161310814 |
| 1 | 0.033159046 | 0.926490464 | 0.073509536 | 10.15130234 | 0.746217523 | 0.116445984 | 0.086894034 |
| 1 | 0.017996989 | 0.959407284 | 0.040592716 | 10.02889493 | 0.407100089 | 0.107886094 | 0.043920438 |
| 1 | 0.00874063 | 0.980075133 | 0.019924867 | 9.954599009 | 0.198344059 | 0.103506704 | 0.02052994 |
| 1 | 0.003798662 | 0.991291399 | 0.008708601 | 9.915066851 | 0.086346362 | 0.101444347 | 0.00875935 |
| 1 | 0.001477283 | 0.996604209 | 0.003395791 | 9.89652979 | 0.033606543 | 0.100560909 | 0.003379504 |
| 1 | 0.000514093 | 0.998816957 | 0.001183043 | 9.888844425 | 0.011698924 | 0.100219425 | 0.001172459 |
| Average | 0.047598199 | 0.901344157 | 0.098655843 | 10.27294017 | 1.049872571 | 0.494294176 | |
| Sum | | | | | Final Cloud Radiance: | | 9.540362663 |

Figure 6-6 CLT vs Step Wise calculations for cloud radiance, part 2 of 3

| | CLT | Step Wise | (SW - CLT) / CLT | % Difference |
|-----------------------------|---------------------|--------------------|---------------------|---------------|
| Total Cloud Radiance | 9.2406943560 | 9.540362663 | -0.032429198 | -3.24% |

Figure 6-7 CLT vs Step Wise calculations for cloud radiance, part 3 of 3

The total cloud radiance CLT value was calculated using Equation 6-10 using the average Gaussian concentration of 0.047598199 [ppm] with an absorption coefficient of 1 [1/ppm-m] and a pathlength of 21 [m]. The average absorption was then used to solve for the average transmission. The average transmission was then used to calculate the average emissivity. The average temperature was then used to calculate the black body radiance (see Equation 2-23). The black body radiance times the average emissivity equals the total CLT cloud radiance.

The stepwise approach calculates the black body radiance and emissivity on a cell by cell basis. The calculations are similar to Equation 6-10 but instead of taking the average value the individual cell value is used. All the individual black body radiances are then summed for a total blackbody radiance. The three last columns of Figure 6-6 illustrate the process.

Figure 6-8 is useful in visualizing the concepts presented by the previous 3 figures.

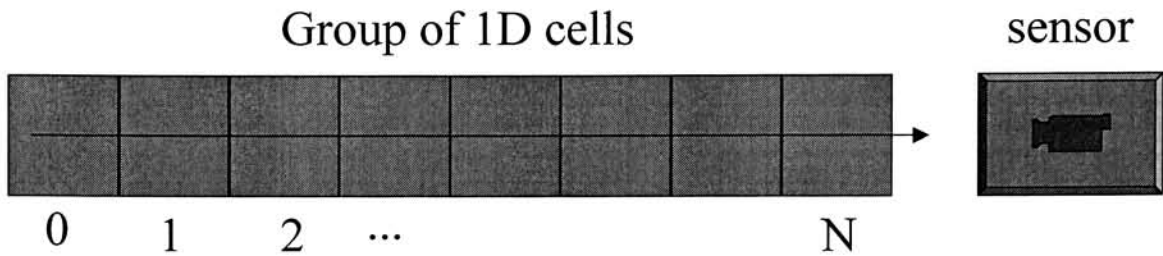


Figure 6-8 Diagram of step wise 1D group of cells

Once the ray leaves the cloud DIRSIG continues with ray tracing through the rest of the scene.

6.3 Noise Artifacts

Noise is defined as an unexpected variation in the signal received at the output. In remote sensing the sources of noise are broken into three major categories: scene, sensor, and processing, see Figure 6-9.

| Noise factors and variability degrading information content | | | | | | | |
|---|----------------------|----------------|-------------------------------|--------------------------|------------------------------|----------------------|-----------------------|
| Scene | | | Sensor | | | Processing | |
| Surface | Goniometric | Atmospheric | Electrical | Mechanical | System | Computational | Analyst |
| -Within class variation | -Illumination angle | -Absorption | -Thermal | -Blur | -Optical transfer function | -Resampling | -Training set size |
| -Background variations | -View Angle | -Scattering | -Shot | -Vibration | -Calibration uncertainty | -Misregistration | -Training set errors |
| -Mixed pixels | -Polarization | -Path radiance | -Quantization | -Multi-sensor distortion | -Scattering within optics | -Compression, coding | -Errors of assumption |
| -Adjacent reflections | -Specular reflection | -Sky radiance | -Crosstalk | | -Scanning & Sampling effects | -Round-off | |
| -Surface slope & aspect | | -Variability | -Temperature induced | | | | |
| -Obscuring objects | | -Cloud effects | -Electrical transfer function | | | | |

Figure 6-9 Noise and information degrading effects in a remote sensing system [Kerekes, 1987]

The remote sensing of gas clouds using multi-spectral simulations would be most effected by spectrally correlated noise. The output digital count is a function of the incident radiance onto the detector. Ideally each detector would output the same digital count for the same incident radiance both spatially and spectrally. Any shift in the system spectral response after ground spectral calibration will cause a shift in the band edges.

If these spectral changes go unnoticed, they can introduce radiometric calibration errors, atmospheric correction errors, and misinterpretation of spectral signatures. The relative importance of these changes will depend a great deal on the calibration techniques used and on where the spectral shift occurs relative to spectral structure in the target or the atmosphere (i.e., a very small change at the edge of an atmospheric window could have a major impact, whereas a large change in the middle of window or in a spectral region where the target and backgrounds were slowly varying would have limited impact.) [Schott, 1997]

The spectral signature of the gas can be altered by spectrally correlated noise. Currently, DIRSIG does not include spectrally correlated noise in the final image. Therefore, spectrally correlated noise should be added in post processing to the final output image in order to investigate the effects on detection of the gas cloud. Unfortunately, spectrally correlated noise is detector dependent. One way to characterize the spectrally correlated noise would be to use dark current readings for each band prior to acquiring an image. The dark current image is the signal generated by the system when there is no incident radiation, thus the dark image can be used to characterize the spatial and spectral noise of the system.

7 Validation

Validation of the gas model requires experimental data. Unfortunately, there is none available at this time. Releases of toxic agents are extremely dangerous and it is for this reason they are not done. The availability of truth data would be a great help in incorporating improvements and investigating the evolution of a cloud in a real environment. An alternative to releasing toxic agents would be to release a benign gas with a similar molecular weight as the toxic gas to be studied. Appendix E briefly describes a design experiment outlining a procedure for such a release. The validation of this research involved verifying offline that DIRSIG was computing the correct values given the equations used. This included sanity checks on mass, volume, total concentration, and spatial sampling as a function of time. Online (inside DIRSIG), phenomenology such as diffusion, reduced concentrations and temperature were investigated. Finally, offline (outside DIRSIG) calculations of integrated radiance, dilution, transmission, column density, and temperature were calculated and compared to the DIRSIG calculations.

7.1 Voxel file validation

Voxel files are created by the C++ program, `make_blob.cc`, and altered to include fBm fields generated by the IDL program `fbm.pro`. Voxel files are used by the DIRSIG gas model and contain information describing the 3D Gaussian PDF. It has the following format:

Line 1: x, y, z insertion points; x, y calculated using wind speed and direction and z calculated using Brigg's buoyancy equation

Line 2: x, y, z Gaussian PDF size; dimensions of Gaussian cube

Line 3: scale - how long each side of a voxel is in gdb units

Line 4 – to EOF: 3D Gaussian PDF in 1D format, where $\text{Gauss}(x,y,z) =$

$z * x_size * y_size + y * y_size + x$; i.e. The value for 3D Gauss(3,5,9) if $x_size = y_size = z_size = 10$ would be found at $900 + 50 + 3 = 953 + 3 =$ line 956 of the voxel file.

The information in these files was checked offline to ensure the total concentration (mass / volume) remains constant throughout a time evolution. This assumes no mass losses due to downwash, dry removal, vegetation, etcetera. Since the mass is conserved the total concentration is a constant mass * inverse volume (represented by the Gaussian PDF). Based on the probability of observations for a normal distribution and using three standard deviations the concentration of a 1D Gaussian with mass of 1 gram is approximately: $1 \text{ [g]} * 0.9974 \text{ [1/m]} = 0.9974 \text{ [g/m]}$. In 3D, the total concentration with mass of 1 gram becomes: $1 \text{ [g]} * (0.9974 \text{ [m]})^3 = 0.9922 \text{ [g/m}^3\text{]}$. The total inverse volume was calculated by summing the individual Gaussian PDF values. The total inverse volume was consistently, out to about four decimal place, around the expected $0.9922 \text{ [1/m}^3\text{]}$. Thus, the concentration is within 0.1 % of the expected value.

7.2 DIRSIG sampling

It is important to determine if DIRSIG samples the voxel file correctly. The sampling size can be determined by knowing the focal length, height above the target, array size, and the scene units. Once the sample size was determined it was then compared to the known voxel dimensions to ensure that the cloud was the proper size. The sample size was determined using similar triangles and the following calculations:

Given: scene units = inches, default x, y array size, $A = 0.974$ [in], image size = 128×128 [pixels], focal length, $f = 0.787$ [in], height above target, $H = 10500$ [in], cloud size = $787 \times 787 \times 787$ [in].

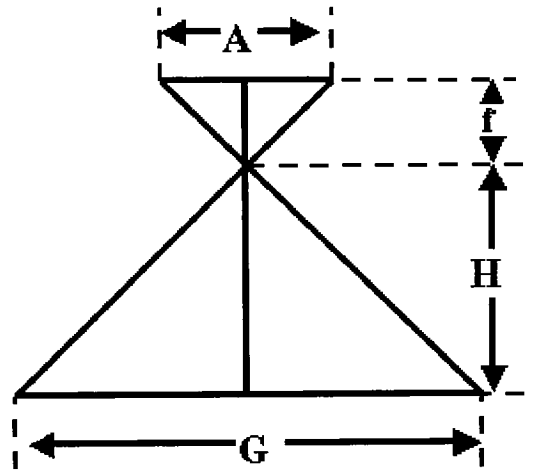


Figure 7-1 Geometric construction for sampling transformation

$$\begin{aligned} \frac{G}{2} [\text{in}] &= \frac{H [\text{in}] * \frac{A}{2} [\text{in}]}{f [\text{in}]} \\ &= \frac{10500 [\text{in}] * 0.487 [\text{in}]}{0.787 [\text{in}]} = 6497.5 [\text{in}] \end{aligned}$$

Equation 7-1

$$\begin{aligned} \text{Detector resolution} [\text{in} / \text{pix}] &= \frac{\frac{G [\text{in}]}{2}}{\frac{\text{image size} [\text{pix}]}{2}} \\ &= \frac{6497.5 [\text{in}]}{64 [\text{pix}]} = 101.5 \left[\frac{\text{in}}{\text{pix}} \right] \end{aligned}$$

Equation 7-2

$$\begin{aligned}
\text{Pixels needed to represent cloud [pix]} &= \frac{\text{Size of cloud [in]}}{\text{Detector resolution } \left[\frac{\text{in}}{\text{pix}}\right]} \\
&= \frac{787 \text{ [in]}}{101.5 \left[\frac{\text{in}}{\text{pix}}\right]} = 7.75 \text{ [pix]}
\end{aligned}$$

Equation 7-3

The number of pixels used by DIRSIG was 8, which is acceptable since discrete sampling methods are used.

7.3 DIRSIG debug images

DIRSIG produces debug images that allow the user to examine various DIRSIG outputs. These include: cloud dilution [$1/\text{m}^3$], cloud temperature [K], cloud column density [ppm-m], cloud transmission, and cloud radiance [$\text{W}/\text{m}^2\Omega$]. It is important that these images contain accurate data based on the equations used. The DIRSIG data was subjected to a comparison offline to validate that the answers were accurate.

Cloud dilution represents the integrated 3D Gaussian values through the cloud at a specified angle. This was tested by constructing a voxel file of known dimensions (20x20x20) with each element set to one. The dilution was calculated offline by integrating through the voxel files and compared to the DIRSIG output. Various sampling scales were used with a nadir look angle. It is important to note that cloud dilution is not a function of area but a function of path length through the cloud. Thus, the dilution should be the same for each pixel regardless of the scale as long as the size of the cloud has not changed. DIRSIG output was within 5-10% of the correct

dilution of 20. The error can be attributed to discrete sampling of the cloud and varies with sampling scale and cloud size. For example, a one-voxel error in the above scenario produces a 5% error. However, that same one voxel error for a voxel file of dimensions 100x100x100 would show only a 1% error. Thus, an error in precision will have a much larger impact for gas clouds that are just starting to evolve.

Cloud temperature calculates the cloud temperature distribution based on the cloud dilution and the initial cloud temperature, see Equation 6-9. These values were calculated offline using the cloud dilution and Equation 6-9 and compared to the DIRSIG output. The DIRSIG calculation was within 5-10% of the correct temperature. Again, these errors arise from errors in the dilution calculations due to discrete sampling.

Cloud column density is the concentration over a given length through the gas based on the cloud dilution, total mass, molecular weight, ideal volume of the gas @ STP, and the total path through the gas cloud (see Equation 6-8). The molecular weight, total mass, and ideal volume are all constants throughout the time evolution. Differences in column density are dependent on the cloud dilution and the look angle. The look angle will determine the path length through the cloud by indicating the vector in and out points for the cloud data. The distance through the cloud, R , is calculated as:

$$R \text{ [m]} = \sqrt{(X_{\text{out}} - X_{\text{in}})^2 + (Y_{\text{out}} - Y_{\text{in}})^2 + (Z_{\text{out}} - Z_{\text{in}})^2}$$

Equation 7-4

Where X_{out} , Y_{out} , Z_{out} are coordinates [m] of the vector leaving the cloud and X_{in} , Y_{in} , Z_{in} are coordinates [m] of the vector entering the cloud.

As expected, using Equation 6-8 the DIRSIG column density was within 5-10% of the calculated column density. Again, the error is attributed to scaling and discrete sampling and will fluctuate based on scale and cloud size.

Cloud transmission is based on the absorption, see Equation 2-13. The absorption is a function of the absorption coefficient, the concentration, and the path length through the cloud, see Equation 2-14. These values were calculated offline using Equation 2-13 and Equation 2-14 and compared to the DIRSIG output. The DIRSIG cloud transmission was within 5-10% of the calculated cloud transmission. Again, the error is attributed to scaling and discrete sampling and will fluctuate based on scale and cloud size.

Cloud radiance [$\text{W/m}^2\Omega$] is based on the emissivity, see Equation 6-3, through the gas cloud times the blackbody radiance, see Equation 2-23. The total radiance was calculated offline and compared to the DIRSIG output. The DIRSIG cloud radiance was within 5-10% of the calculated cloud radiance. The error is attributed to scaling and discrete sampling and will fluctuate based on scale and cloud size.

To summarize, the primary difference between offline calculated values and DIRSIG stems from a discrete sampling issue. The error in precision will have a more profound effect on small gas clouds. It is speculated that the sampling error is introduced because the 3D-DDA does not identify all voxels pierced by the ray. A new sampling algorithm that includes all voxels pierced by a ray may reduce or eliminate this sampling issue, see Heckbert (1994).

8 DIRSIG synthetic scenes

The following section includes images to qualitatively investigate the cloud model.

8.1 Examples

The following animation was rendered with DIRSIG using the theory presented in the previous sections. The pictures illustrate a desert scene with a Soman (Gd) gas cloud and an initial mass of 1 kg. The time duration is 100 – 600 seconds after time $t = 0$ in 100 second increments. The wind speed is 0.5 m/s at a 60 degree angle (90 is due north, 0 is east). The Pasquill stability class is D (neutral) with rural coefficients. The initial gas temperature is 350 K and the spectral bands are integrated over $900 - 1100 \text{ cm}^{-1}$, which is the peak absorption range for Soman gas. Gain and Bias are the same for each image for comparison purposes. It illustrates some interesting phenomenology. The desert sand is very warm due to thermal loading from the sun. The gas cloud is actually cooler than the background. This is why the cloud is darker than the sand. However, compared to the horizon the gas cloud is still warmer than the air. It also illustrates the diffusion of the cloud over time.

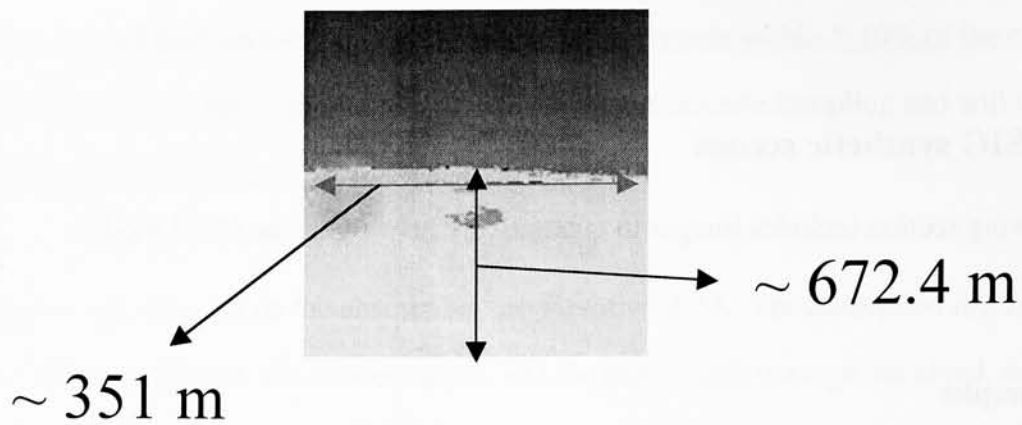


Figure 8-1 X,Y scale in meters

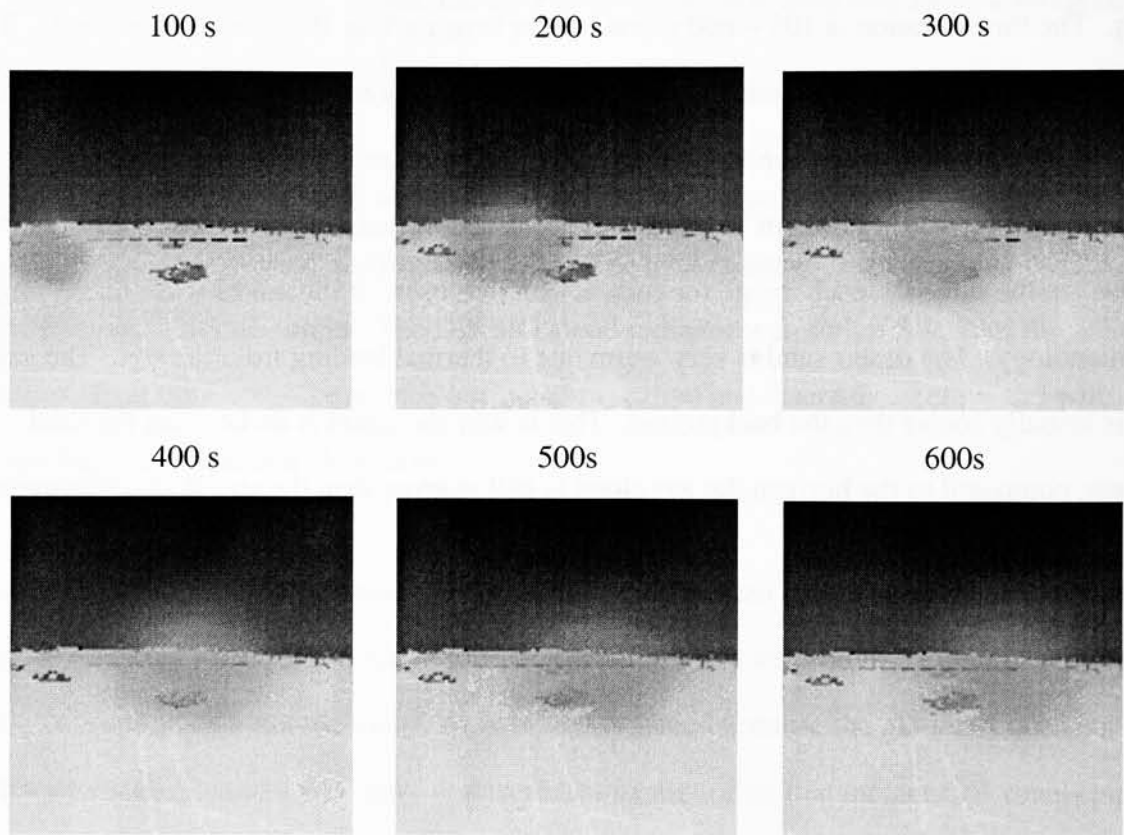


Figure 8-2 Desert animation sequence

The next sequence of images illustrates emission and absorption spectral characteristics of various points in the scene as indicated by the cross hairs on each image. The pictures illustrate

a desert scene with a Soman (Gd) gas cloud and an initial mass of 1 kg. The time duration is 600 seconds after time $t = 0$. The wind speed is 0.5 m/s at a 60 degree angle (90 is due north, 0 is east). The Pasquill stability class is D (neutral) with rural coefficients. The initial gas temperature is 350 K. Same scale as previous images.

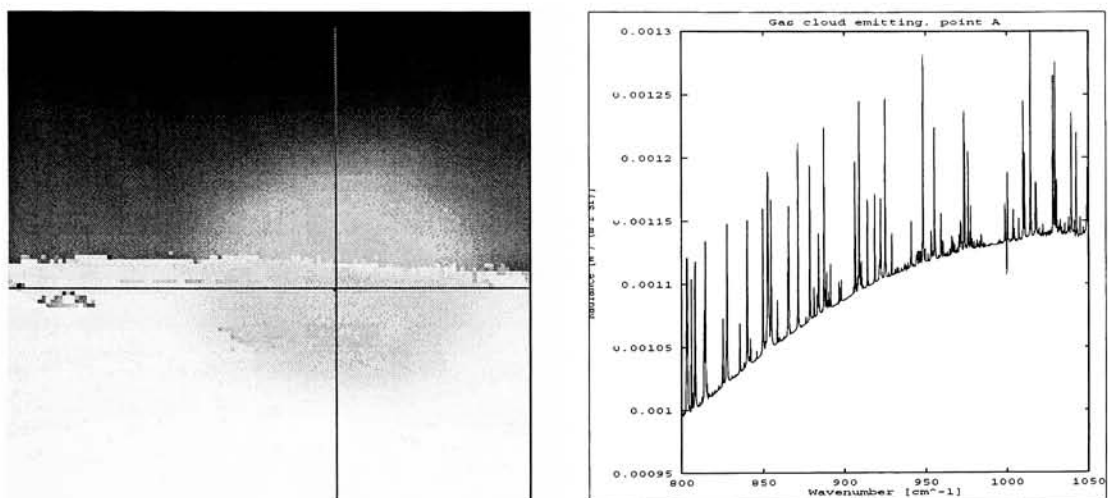


Figure 8-3 Point A, cloud emission with background interaction

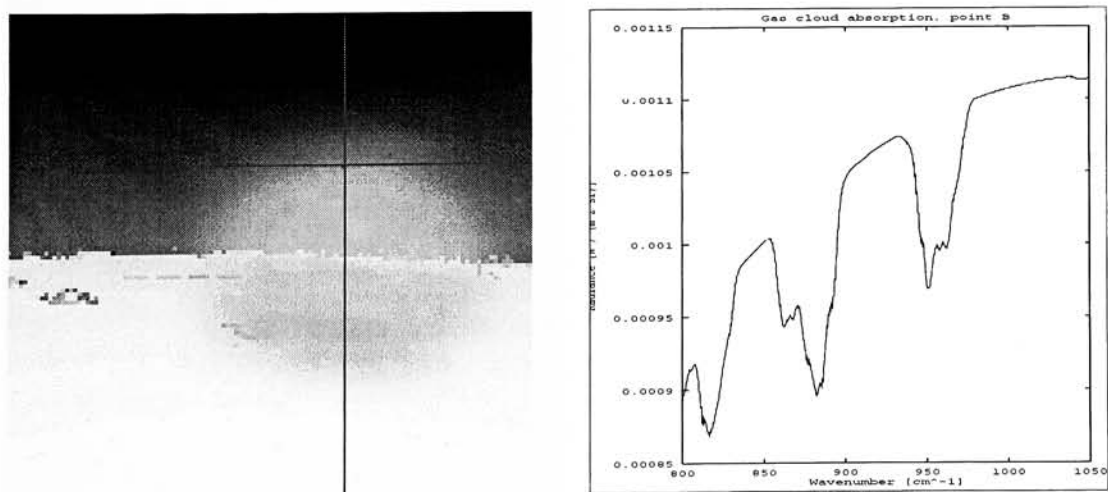


Figure 8-4 Point B, Gas cloud absorbing against atmosphere

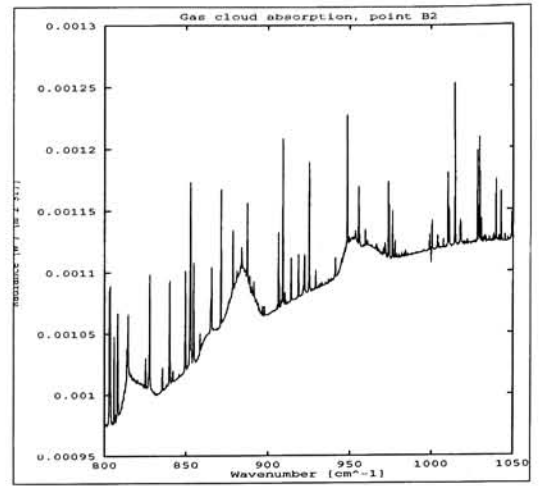
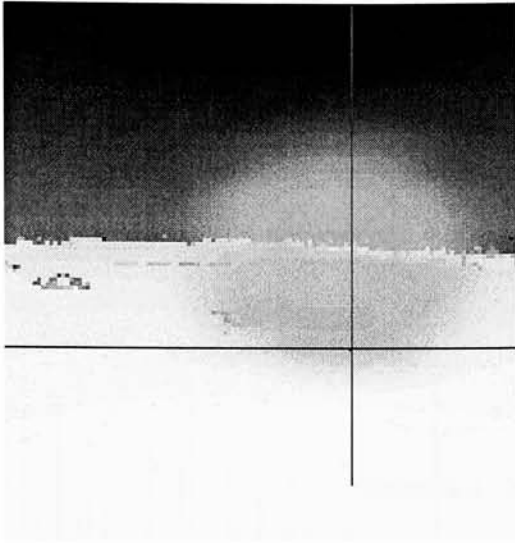


Figure 8-5 Point B2, Gas cloud emission against background

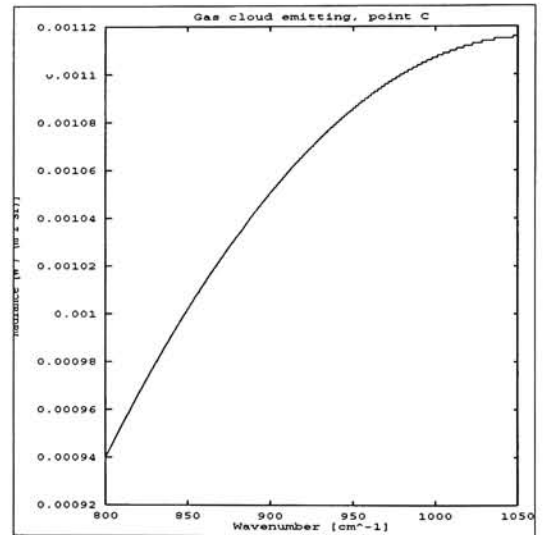
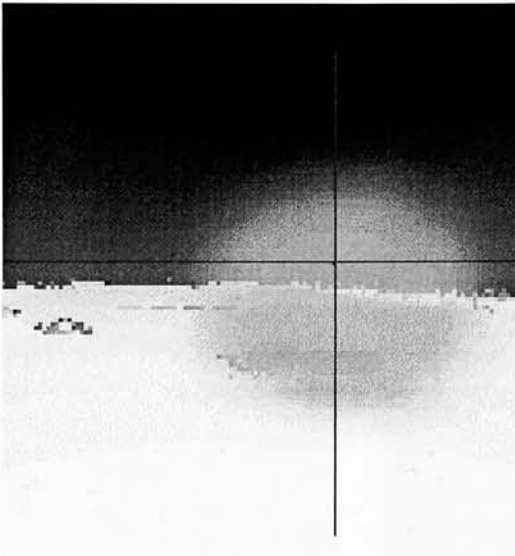


Figure 8-6 Point C, high cloud emission

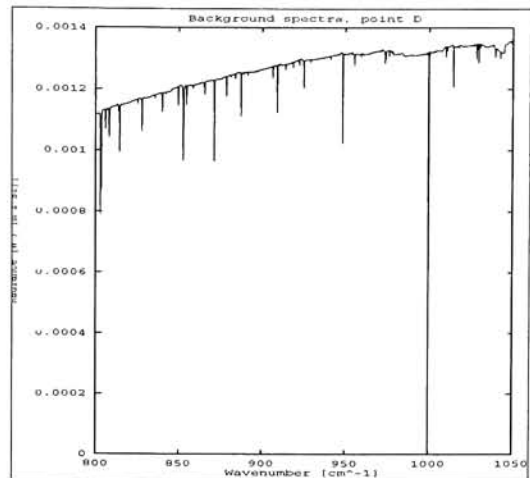
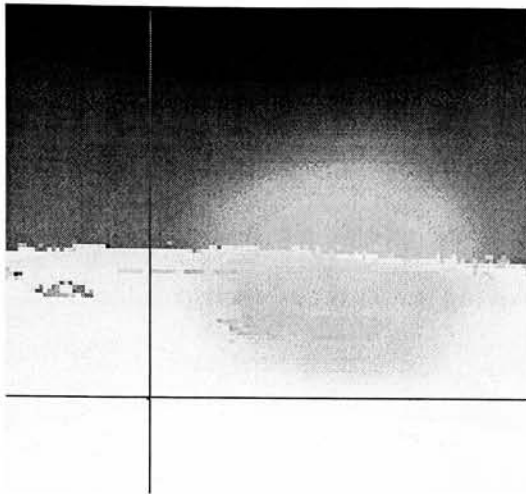


Figure 8-7 Point D, background spectra

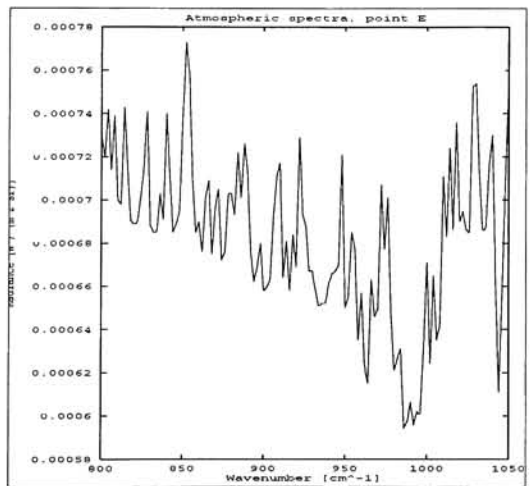
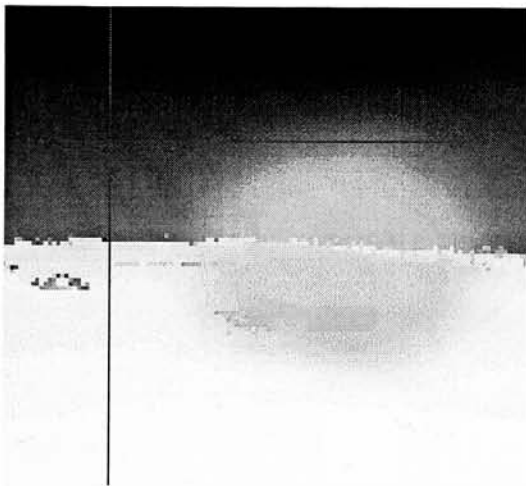


Figure 8-8 Point E, atmospheric spectra

The next pictures are of a scene with a slant angle of 5 degrees at 1500 seconds after release. The initial mass was 200 kg, initial temperature was 350 K and the distance from the cloud to the sensor is approximately 1.5 km. It demonstrates the different absorption peaks at the three wave numbers. It also demonstrates that Soman gas has a higher absorbance in band 1020 $[\text{cm}^{-1}]$ than in bands 900 and 1100 $[\text{cm}^{-1}]$ with band 900 $[\text{cm}^{-1}]$ having the least absorbance of the three. This agrees with the gas absorption data supplied by the Army's Aberdeen Proving Grounds.

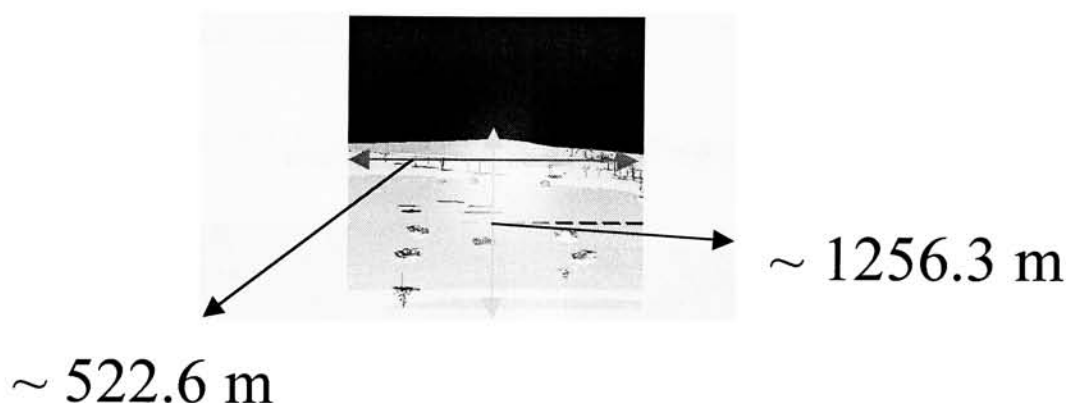


Figure 8-9 X,Y image scale in meters

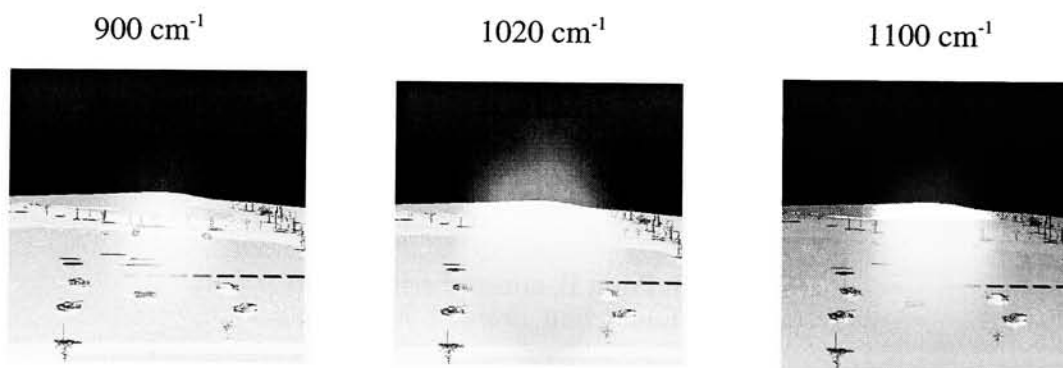


Figure 8-10 Desert release, 1500 seconds

The next sequence of images illustrate an urban scene with a Soman (Gd) gas cloud and an initial mass of 0.5 kg. The time duration is 25 500 seconds in 100 second increments after time $t > 0$.

The wind speed is 0.5 m/s at an 80 degree angle (90 is due north, 0 is east). The Pasquill stability class is A (unstable) with urban coefficients. The initial gas temperature is 350 K and the spectral bands are integrated over 990 - 1100 $[\text{cm}^{-1}]$ in 10 $[\text{cm}^{-1}]$ increments. Gain and Bias are the same for each image for comparison purposes.

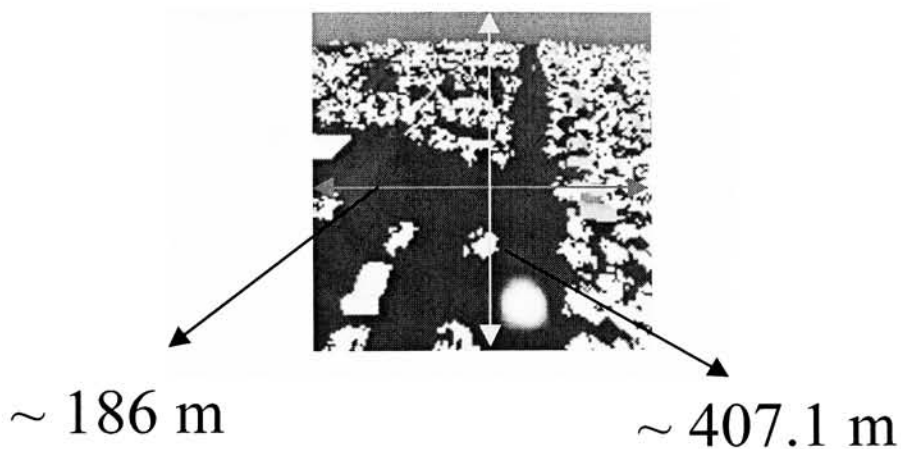
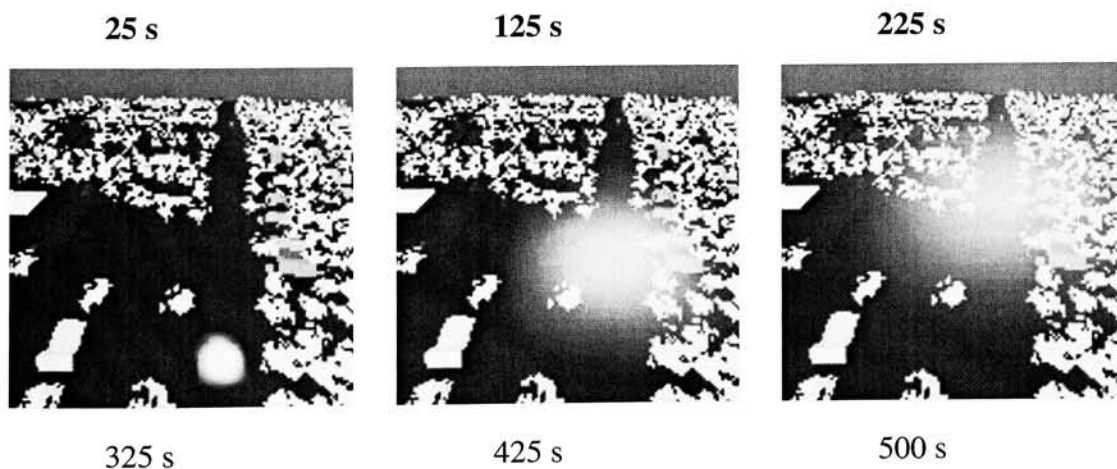


Figure 8-11 X,Y scale in meters



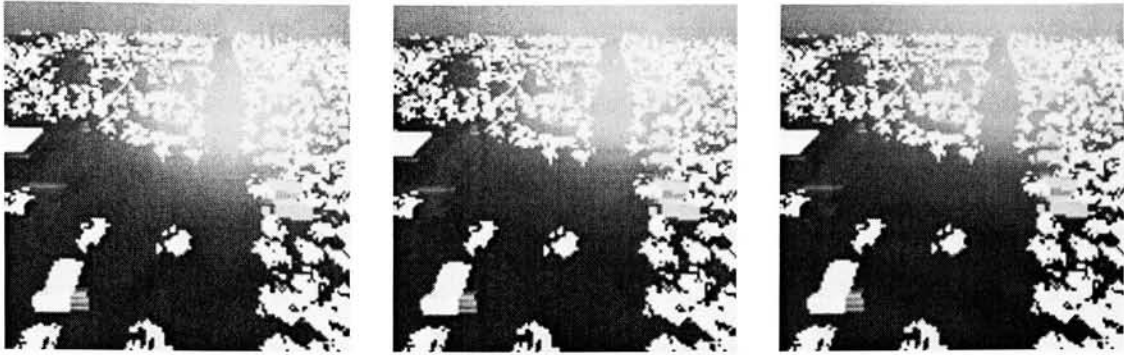


Figure 8-12 Urban animation sequence

The animations shown above demonstrate a decrease in concentration and emissivity as the size of the gas cloud increases. This follows the theoretical expectations.

The next set of images demonstrates the spectral “finger print” of the cloud as a function of transmission through the cloud. The scene shows a 1 Kg release at 575 seconds after $t = 0$. The Pasquill stability class is D (neutral) with rural coefficients. The initial gas temperature is 350 K and the spectral range is $800 - 1050 \text{ [cm}^{-1}\text{]}$ at $0.25 \text{ [cm}^{-1}\text{]}$ increments (FASCODE). The background is a black body radiator with temperature set to 0 K. For these scenes the atmosphere between the target and sensor has been removed. Same image scale as Figure 8-1.

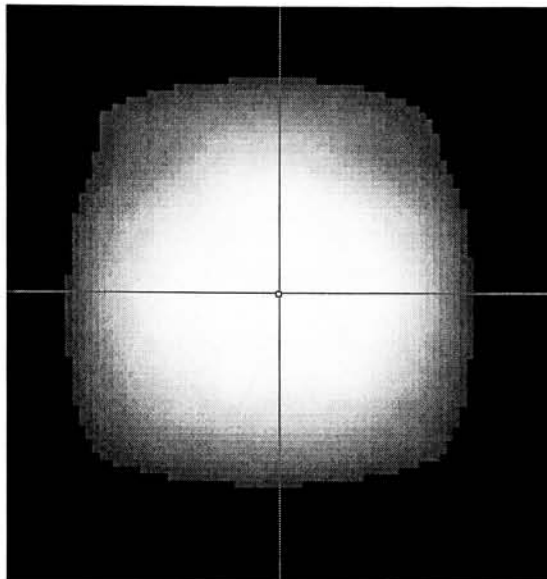
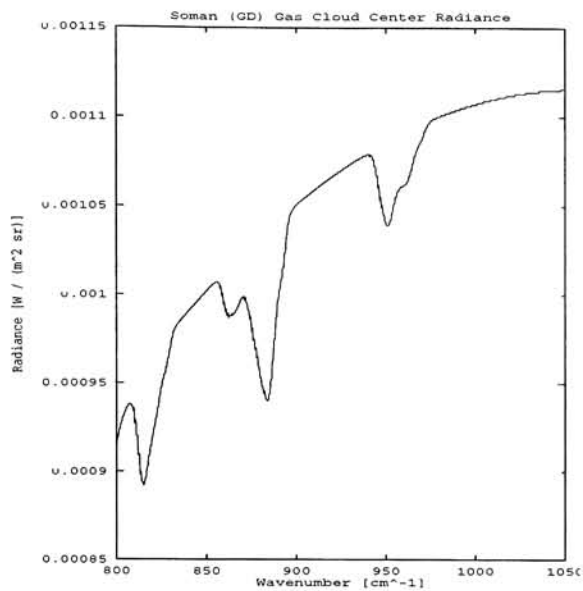


Figure 8-13 Spectral "finger print" radiance for center of GD gas cloud

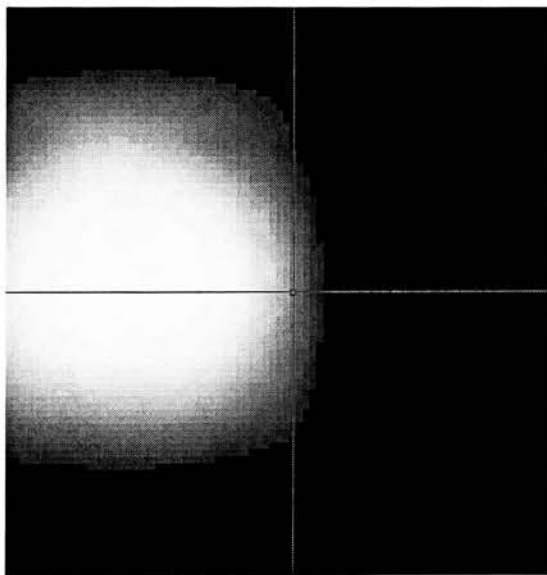
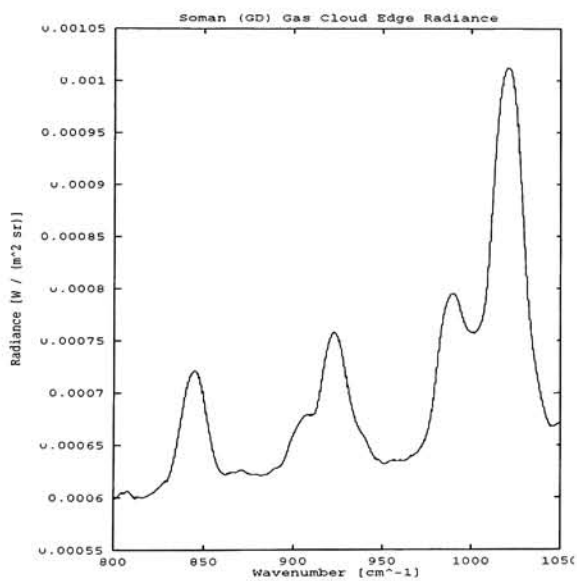


Figure 8-14 Spectral "finger print" radiance for edge of GD gas cloud

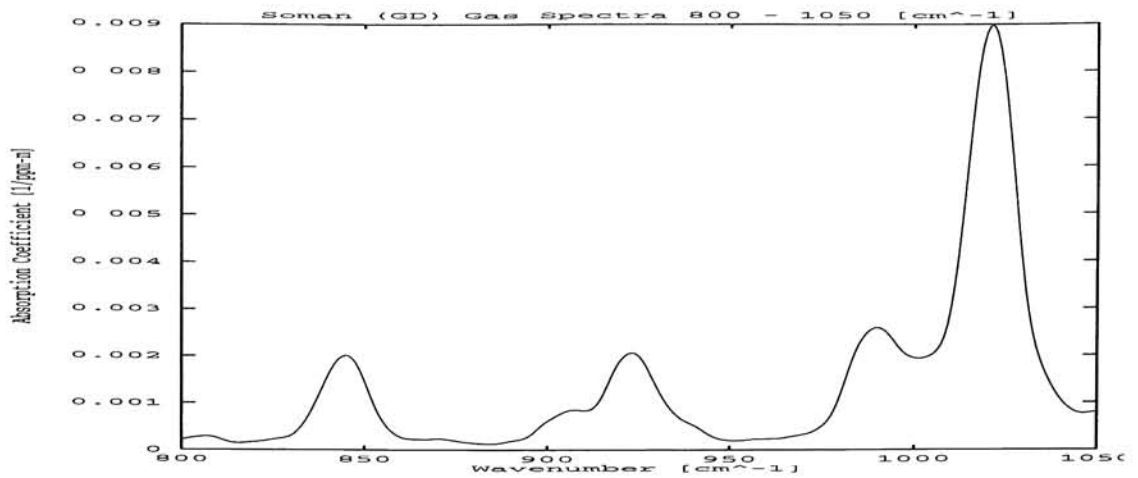


Figure 8-15 Absorption coefficients of Soman (GD) 800 - 1050 [cm⁻¹]

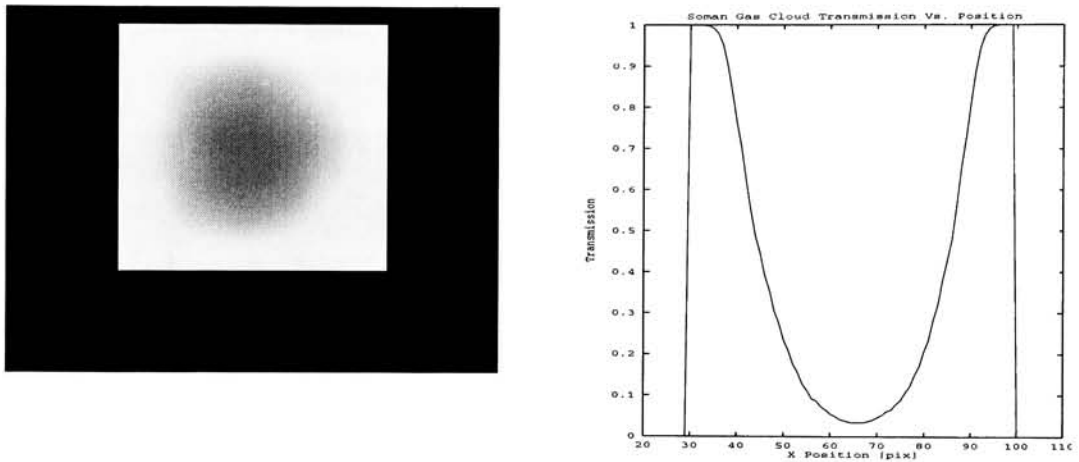


Figure 8-16 Soman (GD) gas cloud transmission for Figure 8-13 and Figure 8-14

As expected, see Equation 6-10, the images above demonstrate that when the transmission is low, i.e. the center of the cloud, the spectral “finger print” approaches a black body radiator. When the transmission is high, i.e. at the cloud edge, the spectral “finger print” takes on the shape of the absorption coefficients.

The following images illustrate background interaction, atmospheric absorption due to water lines and other gases, and release mass differences for a Soman (GD) release 1500 seconds later, wind speed = 1 m/s, Pasquill class A, and rural coefficients. This is a FASCODE run with a wavenumber increment of 0.25 from 800 – 1050 [cm^{-1}]. The scale is shown in Figure 8-9.

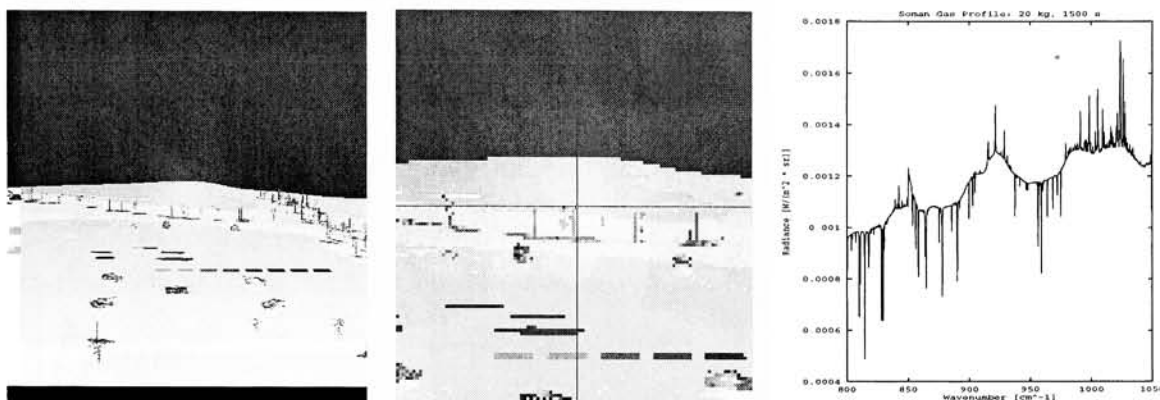


Figure 8-17 Soman 20 kg release background: original, zoom, and spectral profile for location indicated by crosshairs

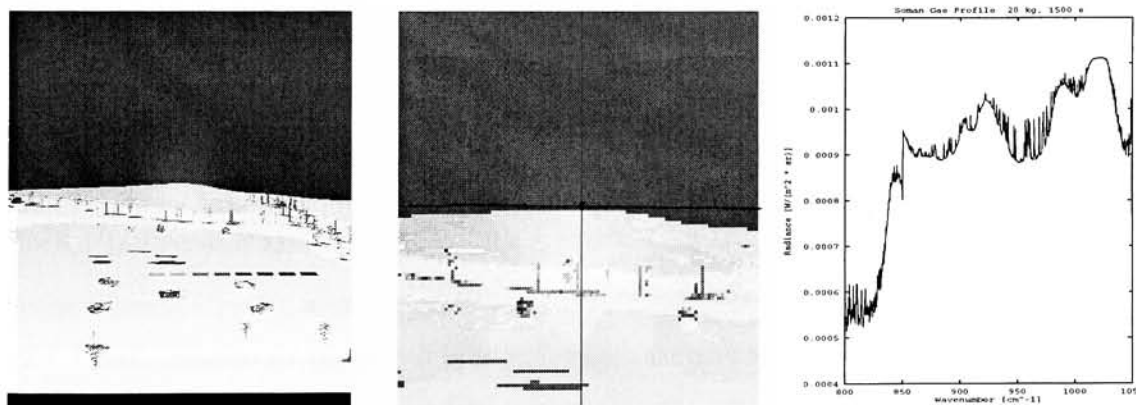


Figure 8-18 Soman 20 kg release horizon: original, zoom, and spectral profile for location indicated by crosshairs (taken just above horizon)

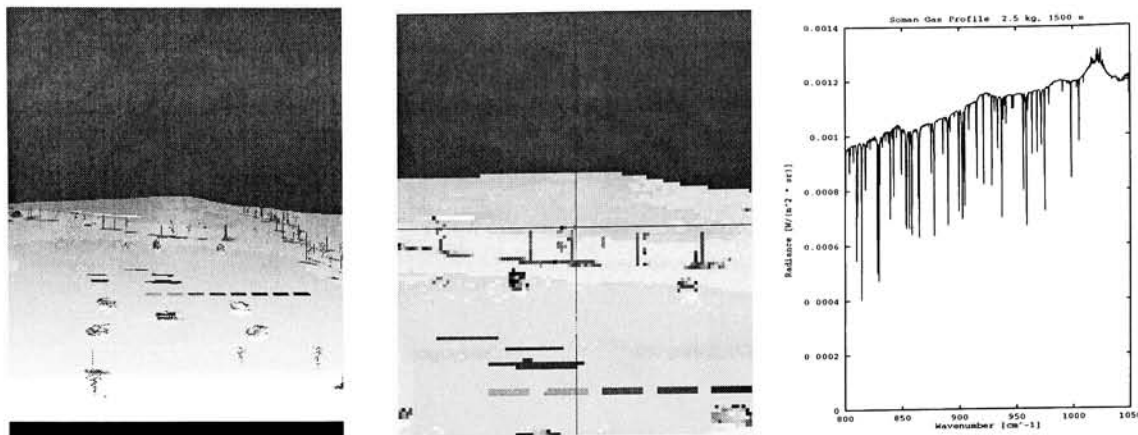


Figure 8-19 Soman 2.5 kg release background: original, zoom, spectral profile for location indicated by crosshairs

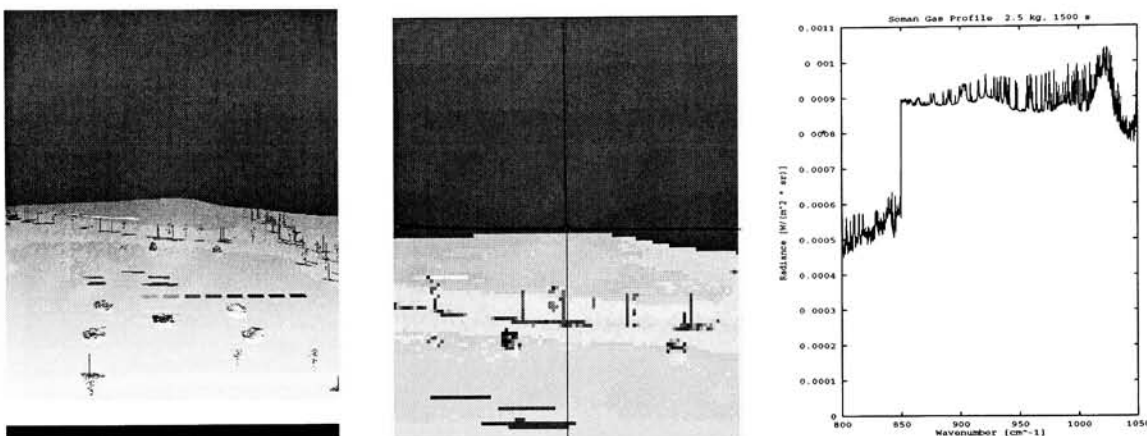


Figure 8-20 Soman 2.5 kg release horizon: original, zoom, spectral profile for location indicated by crosshairs (taken just above horizon)

The Soman peaks are clearly visible against the background for the 20-kg release. The 2.5-kg release is much more difficult to resolve. The water absorption lines and background spectra coupled with the smaller release mass make detection a much harder task. The concentration is still at a toxic level and exposure upwards of 1 hour could cause death. Detection algorithms would be best suited for this type of release where concentrations are small but still hazardous to human life. It is important to note that these spectra do not contain spectrally correlated noise. Detection of these low concentrations with added spectral noise may prove to be very difficult.

9 Conclusion and Improvements

Due to the complex nature of this research it was difficult to acquire specific references to modeling and simulating chemical weapons dispersal patterns. The few references found specifically on the subject are very vague in nature. They do not discuss specific dispersion equations, weather interactions, or methods of dissemination (i.e., explosive models, airflow, etc...). This drove the direction of this research to use theory based on the extensive studies and papers explaining factory stack plume dispersion.

The cloud model presented is a generic Gaussian model that can use any gas absorption spectra. It applies first principle physics to simulate drift and dispersion. The dispersion and drift equations are based on experimental data obtained from factory plumes. These equations incorporate the concepts of turbulence and buoyancy effects. The model also encompasses fractal fBm techniques to reproduce micro-scale variances within the cloud. The cloud was successfully integrated into DIRSIG and rendered using MODTRAN, FASCODE, and basic radiometric principles. This allows the user to produce ultra-spectral data cubes to test detection and quantification algorithms without the need for expensive and dangerous field releases. The model was also used to illustrate common phenomenology such as diffusion and contrast reversal. In addition, it allows the user to produce animations to represent the evolution of the cloud over time. Finally, DIRSIG images of the integrated column density, temperature, transmission, and radiance are included as supplementary data and can be used as “truth” for algorithm development.

The model presented does not include all the complex phenomenology that would effect the evolution of a gas' cloud. It is meant as a simple model that makes assumptions that may or may not be valid depending on the situation (I.e. no wet/dry removal, no wind variation, conservation of mass, etc...). It is a starting point from which to build upon. There is no experimental data for gaseous nerve agents. Hence, there is no complete way to validate the accuracy of this model. However, a benign gas could be used as an alternative to study dispersion characteristics. The scenes rendered are radiometrically sound based on the exit temperature and PDFs used for the gas cloud. However, it is a common phrase, that garbage in produces garbage out. If the PDF assumption and dispersion coefficients were incorrect for these nerve agents then the rendered output would be questionable.

Improvements to the model in order of importance:

1. Collection of truth data of a gas dissemination to validate model (see Appendix E).
2. Provide interaction with complex terrain. This would include the calculation of the flow (wind and temperature) around various complex objects. I.e. buildings (downwash), hills, valleys (drainage flow), vehicles, land/water interface (over water and coastal dispersion), foliage, etc...
3. As discussed in section 6.2.1, include step wise vs central limit theorem cloud calculations
4. Blast kinetics and how it relates to the cloud PDF.
5. A 3D-DDA that includes all pixels pierced by a ray. The pixels should be weighted between 0 and 1. I.e., a direct hit of a pixel would be one and all other hits would be less then one. One idea is to weight the pixel by the length of the vector that pierces the pixel compared to the length of the vector from opposite vertex to opposite vertex. A hit perpendicular to the face of a voxel would be a special case. This should make

the cloud appear less voxelized or more continuous. Incorporate the dimension of time in the fBm texture map. This would permit the simulation of complex turbulent behavior. In addition, the temporal correlation function could be used to speed up the rendering of subsequent frames in a time sequence.

6. Include the Komolgorov microscale for small scale turbulence. See Sakas (1993).
7. As discussed in Appendix D , include spectrally correlated noise.
8. The effects of temperature/pressure broadening on spectral signatures.
9. Wet/dry removal (mass depletion) based on collisions with the ground or other objects or removal due to precipitation. Plume models listed by the EPA often include wet/dry removal. This would be a good starting point for information².
10. Possible precipitation or other chemical transformations of the gas downwind.
11. Seasonal variations. I.e. winter, summer, fall, and spring.

² <http://www.epa.gov/epahome/models.htm>

10 Appendix A: Voxel Creation Code

The actual code is not included here in order to save space. The code will be included in an accompanying CD-ROM.

This section discusses the C++ program `make_blob.cc` and the IDL program `fbm.pro`. The two programs are used in conjunction to create fBm textured gas clouds.

10.1 `Make_blob.cc` and `blob.dat`

`Make_blob.cc` is a program written in C++ based on the experimental data and equations presented by Pasquill and Briggs in the section 4. Pasquill discusses how to determine the standard deviation to be used in the Gaussian equation. Briggs examines how to determine the rise associated with a cloud based on initial momentum and temperature. Both the standard deviation and rise are associated with Pasquill classes. These Pasquill classes incorporate the effect of turbulence on the cloud. `Make_blob.cc` reads in a data file which describes the environmental conditions (i.e., wind speed, insolation, rate of release, cloud cover, starting x, y, z coordinates, etc...). It then uses this information to determine the Pasquill class, standard deviation of the Gaussian PDF, and the rise (based on the Brigg's bent, over hot plume equations). `Make_blob.cc` then outputs a Gaussian PDF and final x, y, z coordinates according to the Pasquill and Brigg's equations. The voxel file has the following format:

Line 1: x, y, z insertion points; x, y calculated using wind speed and direction and z calculated using Brigg's buoyancy equation

Line 2: x, y, z Gaussian PDF size; dimensions of Gaussian cube

Line 3: scale - how long each side of a voxel is in gdb units

Line 4: used to adjust total volume to volume dictated by number of standard deviations.

It is a function of the voxel scale.

Line 5 – to EOF: 3D Gaussian PDF in 1D format, where $\text{Gauss}(x,y,z) =$

$z * x_size * y_size + y * y_size + x$; i.e. The value for 3D Gauss(3,5,9) if $x_size = y_size = z_size = 10$ would be found at $900 + 50 + 3 = 953 + 43 =$ line 957 of the voxel file.

This format is not recognized by DIRSIG. Fbm.pro reads this format and outputs the proper DIRSIG format. To make_blob.cc output DIRSIG readable format:

Comment out line 331 and replace with:

```
Fprintf( fp_out, "%d %d %d\n",  
(int)(stdev*x_size/vxl_scale),(int)(stdev*y_size/vxl_scale),(int)(stdev*z_size/vxl_scale));
```

comment out line 381

Blob.dat is a text file containing three data lines describing environmental and other pertinent parameters. The data lines start with the letters **d**(ata), **p**(osition), or **b**(uoyancy). Make_blob.cc reads the data in blob.dat to create the Gaussian blobs. The following illustration is an example of a blob.dat file:

data file for make_blob.cc, # means a comment follows:

#data parameters

wind [m/s], cloud [0-none, 1-all cloud], insolation [w/m^2],

am_pm [0-am, 1-pm], time [s], rur_urb [0-rur, 1-urb],

Pclass [A-F,G,H,I,Z - let model choose based on wind & insolation],

time increment [divisions from time =0 to time end]

#position parameters

#start_x, start_y, start_z, theta [degrees 0-360], scale [dist/voxel], units

#[F = feet, M = meters, C = centimeters, I = inches], vxl_scale

#buoyancy parameters

#exit_vel [m/s], exit_radius [m], temp_air [K], temp_gas [K]

d 0.5 0 600 1 0 600 1 Z 1

p 38292 33760 3000 60 1 1 2.0

b 1.0 .2 273 300

where

1. wind = wind speed in meters per second
2. cloud = percentage of cloud cover, range is from 0 - 1. 0 is no clouds, 1 is 100%
3. insolation = amount of irradiance in watts per meter squared the gas encounters
4. am_pm = night or day, 0 - day, 1 - night
5. start_time = start time of vxl creation in seconds
6. end_time = end time after release in seconds
7. rur_urb = 0 - rural dispersion coefficients used, 1 - urban dispersion coefficients used
8. Pclass Pasquill class, A-F, G = average of A + B, H = Average of B + C, I = Average of C + D. set equal to Z to let make_blob calculate based on wind speed, insolation, and day/night. Must be in CAPS.
9. time increment = number of increments to break time up into. I.e. if time was set to 100 seconds and you wanted a vxl_file every 10 seconds then the increment would be 10.
10. start_x, start_y, start_z - x, y, z starting coordinates
11. theta = wind direction in degrees, 0 = due east, 90 = due north.
12. scale = set to 1
13. units = gdb units used by DIRSIG: C - cm, M - m, I - inches, F - feet, must

be in CAPS.

14. `vx1_scale` = sampling scale to use when creating `vx1_file`. I.e., 0.5 = 2 pixels for every one GDB_UNIT, 1 = 1 pixel for every GDB_UNIT, 2 = 0.5 pixels for every GDB_UNIT, etc...smaller than one is oversampling, greater than one is undersampling. Allows the creation of very large Gaussian blobs through down sampling or very small Gaussian blobs through over sampling.
15. `exit_vel` = rate of gas from dissemination unit in meters per second
16. `exit_radius` = diameter of dissemination unit, set small for a point source release, set larger for a larger object release
17. `temp_air` = ambient air temperature in Kelvin at the time of release
18. `temp_gas` = temperature of gas in Kelvin at release

To create an animation change start time and end time to cover the time window of interest. Set `time_increment` to break up range into divisions. Set all other parameters. Run `make_blob` executable. I.e., to make an animation from 25 seconds to 225 seconds after release at 10 second intervals set start time = 15 , end time = 225 and `time_inc` = 21. This will produce `blob??vx1`. The question marks will be the time in seconds after release. Each time increment will have its own blob file. ALL PARAMETERS MUST BE SET!!!. `Make_blob.cc` does not provide error checking so it may fail or produce erroneous results if all parameters are not set.

10.2 Fbm.pro

`Fbm.pro` is an interactive display language (IDL) program used to create texture maps for each voxel file created by `make_blob.cc`. The source code requires the language IDL to run. `Fbm.pro` requires the modules: `gaus_amp.pro`, `gaus_phase.pro`, and `gaus_randist.pro`. `Gaus_amp.pro`

generates the amplitudes required for fractional Brownian motion. Gaus_phase.pro generates the Hermetian phase distribution. Gaus_randist.pro generates a 3D random Gaussian distribution with even symmetry about its center. The program is based on the theory in section 3.5.1.

To run fBm.pro move all the voxel files into the same directory as fBm.pro. Run fBm.pro in IDL by typing:

```
>idl  
idl>.run fBm.pro  
idl>fBm
```

The output will be files called fBm_blob???.vxl. The question marks are the time in seconds after the initial release.

The parameter B can be changed in the file: gaus_amp.pro line 28.

It is currently set to 1.50. This will change the weighting of frequencies included in the $1 / f^B$ relationship. Decreasing this number will add more noise.

The variance can be changed on line 12 of fBm.pro. This controls the apparent randomness around the mean.

The fBm voxel file has the following format:

Line 1: x, y, z insertion points; x, y calculated using wind speed and direction and z calculated using Brigg's buoyancy equation

Line 2: x, y, z Gaussian PDF size; dimensions of Gaussian cube

Line 3: scale - how long each side of a voxel is in gdb units

Line 4 – to EOF: 3D Gaussian fBm PDF in 1D format, where $\text{Gauss}(x,y,z) =$

$z * x_size * y_size + y * y_size + x$; i.e. The value for 3D Gauss(3,5,9) if $x_size = y_size =$

$z_size = 10$ would be found at $900 + 50 + 3 = 953 + 3 =$ line 956 of the voxel file.

This is the format required by DIRSIG.

11 Appendix B: DIRSIG Configuration File Changes

This section discusses the configuration changes (extension .cfg) needed by DIRSIG in order to run the gas model. See Brown (1999) for a complete description of each section.

11.1 Original Configuration File Example

This section includes an example configuration file required for DIRSIG runs. It is also required by make_adb to create the atmospheric database (extension .adb).

```
DIRSIG_CFG
#
# DIRSIG Run-Time Configuration File
#

PATHS {
    EMISSIVITY_PATH = /dirs/home/dirsig/training/desert/emissivity
    EXTINCTION_PATH = /dirs/home/dirsig/training/desert/extinction
    MAPS_PATH = /dirs/home/dirsig/training/desert/maps
}

#
# The SCENE section describes what geometric database(s) are being
# modeled, what units the GDB is in, what altitude a height of 0.0 in
# the GDB translates to, where the scene is physically located, time
# of day, day of year,
#
SCENE {
    GDB_FILENAME = ./desert.gdb
    GDB_UNITS = INCHES
    MATERIAL_FILENAME = ./desert.mat
    GROUND_ALTITUDE = 0.218
    DATE = 6 24 1996
    GMT_TIME = 17.50
    GMT_OFFSET = 5.0
    LATITUDE = 43.00
    LONGITUDE = 78.00
```

```

}

#
# The ENVIRONMENT section describes the atmosphere and meteorological
# conditions at the scene.
#
ENVIRONMENT {
    TAPE5_FILENAME = ./desert.tp5
    ADB_FILENAME = ./desert_nadir.adb
    WEATHER_FILENAME = ./desert.wth
}

#
# The PLATFORM section describes the sensor system being used for the
# simulation. This includes information such as the position of the
# sensor and the geometric/spectral description of each band in the
# system.
#
PLATFORM {

    POSITION {
#     TARGET_LOCATION = 1840.00, 1840.00, 0.0
#     TARGET_LOCATION = 960.00, 1216.00, 0.0
#     PLATFORM_LOCATION = 960.00, 2000.00, 794.0
        TARGET_LOCATION = 41894.0, 39987.0, 3000.0
        PLATFORM_LOCATION = 41894.0, 39987.0, 13500.0

        AZIMUTH_ANGLE = 90.0
#     ELEVATION_ANGLE = 70.0
    }

    INSTRUMENT {
        NAME = Generic FTS Imager
        TYPE = FRAMING_ARRAY
        FOCAL_LENGTH = 20.00

        BAND_LIST {

            BAND {

#             MINIMUM_FREQUENCY = 850.0
                MINIMUM_FREQUENCY = 900.0
#             MAXIMUM_FREQUENCY = 1100.0
                MAXIMUM_FREQUENCY = 1100.0
                DELTA_FREQUENCY = 10.0
                RESPONSE_FILENAME = SPECTRAL
            }
        }
    }
}

```

```

        IMAGE_FILENAME = blob_desert_nadir.img

        X_PIXELS = 256
        Y_PIXELS = 256
    }

}

}

}

# flags for optional modeling
OPTIONS {
    ENABLE_TRUTH_IMAGES = TRUE
    ENABLE_MAPS = TRUE
    ENABLE_THERMAL_MODEL = TRUE
    ENABLE_PLUMES = TRUE
}

#
# The TRUTH_IMAGES section controls which debug images will be generated.
# If this section does not appear, the default debug list in
# $DIRSIG_HOME/lib/default.dbl is used. This section is ignored if the
# ENABLE_TRUTH_IMAGES flag is set to FALSE.
#
TRUTH_IMAGES {
    IMAGE_FILENAME = blob_desert_nadir_truth.img
    MATERIAL_MAPS = TRUE
    HIT_MAPS = TRUE
    TEMPERATURE_MAPS = TRUE
    PLUME_MAPS = TRUE
}

#
# The MAPS section describes material and texture maps applied to
# various materials in the scene. This section is ignored if the
# ENABLE_MAPS flag is set to FALSE.
#
MAPS {

    # override desert_wash areas with scene map
    MATERIAL_MAP {
        IMAGE_FILENAME = desert_map.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 26.6
        MATERIAL_ID = 100
        LUT {

```

```

        0 = 8 # map desert_wash (8) to 0
        128 = 9 # map dirt_road (9) to 128
        255 = 10 # map desert_pavement (10) to 255
    }
}

# Texture for sand
TEXTURE_MAP {
    IMAGE_FILENAME = sand.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 15.0
    MATERIAL_ID = 8
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}

# Texture for dirt road
TEXTURE_MAP {
    IMAGE_FILENAME = dirt.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 15.0
    MATERIAL_ID = 9
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}

# Texture for desert pavement
TEXTURE_MAP {
    IMAGE_FILENAME = dp.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 15.0
    MATERIAL_ID = 10
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}

# Texture for desert bushes
TEXTURE_MAP {
    IMAGE_FILENAME = gaussian.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 1.0
    MATERIAL_ID = 28
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}

```

```

# Texture for deciduous #1
TEXTURE_MAP {
    IMAGE_FILENAME = gaussian.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 1.0
    MATERIAL_ID = 31
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}

```

```

# Texture for deciduous #2
TEXTURE_MAP {
    IMAGE_FILENAME = gaussian.pgm
    INSERT_POINT = 25000,75000,0
    GSD = 1.0
    MATERIAL_ID = 33
    MINIMUM_WAVELENGTH = 0.45
    MAXIMUM_WAVELENGTH = 0.52
}
}

```

11.2 Gas Model Configuration File Example

BOLD text indicates sections that need to be included to incorporate the gas model into DIRSIG. The major difference is the inclusion of a **PLUME** section located at the end of the configuration file. Make_adb does not recognize the **PLUME** section. When creating the atmospheric database with make_adb the user must leave out the **PLUME** section.

```

DIRSIG_CFG
#
# DIRSIG Run-Time Configuration File
#

PATHS {
    EMISSIVITY_PATH = ./ems
    EXTINCTION_PATH = ./ext
    MAPS_PATH = ./maps
    ABSORPTION_PATH = ../abs_data
}

#

```

```

# The SCENE section describes what geometric database(s) are being
# modeled, what units the GDB is in, what altitude a height of 0.0 in
# the GDB translates to, where the scene is physically located, time
# of day, day of year,
#
SCENE {
    GDB_FILENAME = ./desert.gdb
    GDB_UNITS = INCHES
    MATERIAL_FILENAME = ./desert.mat
    GROUND_ALTITUDE = 0.218
    DATE = 6 24 1996
    GMT_TIME = 17.50
    GMT_OFFSET = 5.0
    LATITUDE = 43.00
    LONGITUDE = 78.00
}

#
# The ENVIRONMENT section describes the atmosphere and meteorological
# conditions at the scene.
#
ENVIRONMENT {
    TAPE5_FILENAME = ./desert.tp5
    ADB_FILENAME = ./desert_nadir.adb
    WEATHER_FILENAME = ./desert.wth
}

#
# The PLATFORM section describes the sensor system being used for the
# simulation. This includes information such as the position of the
# sensor and the geometric/spectral description of each band in the
# system.
#
PLATFORM {

    POSITION {
#     TARGET_LOCATION = 1840.00, 1840.00, 0.0
#     TARGET_LOCATION = 960.00, 1216.00, 0.0
#     PLATFORM_LOCATION = 960.00, 2000.00, 794.0
        TARGET_LOCATION = 41894.0, 39987.0, 3000.0
        PLATFORM_LOCATION = 41894.0, 39987.0, 13500.0

        AZIMUTH_ANGLE = 90.0
#     ELEVATION_ANGLE = 70.0
    }
}

```



```

INSTRUMENT {
    NAME = Generic FTS Imager
    TYPE = FRAMING_ARRAY
    FOCAL_LENGTH = 20.00

    BAND_LIST {

        BAND {

            MINIMUM_FREQUENCY = 900.0
            MAXIMUM_FREQUENCY = 1100.0
            DELTA_FREQUENCY = 10.0
            RESPONSE_FILENAME = SPECTRAL
            IMAGE_FILENAME = fbm_blob100_desert_nadir.img

            X_PIXELS = 128
            Y_PIXELS = 128
        }

    }
}

# flags for optional modeling
OPTIONS {
    ENABLE_TRUTH_IMAGES = TRUE
    ENABLE_MAPS = TRUE
    ENABLE_THERMAL_MODEL = TRUE
    ENABLE_PLUMES = TRUE
}

#
# The TRUTH_IMAGES section controls which debug images will be generated.
# If this section does not appear, the default debug list in
# $DIRSIG_HOME/lib/default.dbl is used. This section is ignored if the
# ENABLE_TRUTH_IMAGES flag is set to FALSE.
#
TRUTH_IMAGES {
    IMAGE_FILENAME = fbm_blob100_desert_nadir_truth.img
    MATERIAL_MAPS = TRUE
    HIT_MAPS = TRUE
    TEMPERATURE_MAPS = TRUE
    PLUME_MAPS = TRUE
}

```

```

#
# The MAPS section describes material and texture maps applied to
# various materials in the scene. This section is ignored if the
# ENABLE_MAPS flag is set to FALSE.
#
MAPS {

    # override desert_wash areas with scene map
    MATERIAL_MAP {
        IMAGE_FILENAME = desert_map.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 26.6
        MATERIAL_ID = 100
        LUT {
            0 = 8 # map desert_wash (8) to 0
            128 = 9 # map dirt_road (9) to 128
            255 = 10 # map desert_pavement (10) to 255
        }
    }

    # Texture for sand
    TEXTURE_MAP {
        IMAGE_FILENAME = sand.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 15.0
        MATERIAL_ID = 8
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }

    # Texture for dirt road
    TEXTURE_MAP {
        IMAGE_FILENAME = dirt.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 15.0
        MATERIAL_ID = 9
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }

    # Texture for desert pavement
    TEXTURE_MAP {
        IMAGE_FILENAME = dp.pgm
        INSERT_POINT = 25000,75000,0
    }

```

```

        GSD = 15.0
        MATERIAL_ID = 10
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }

    # Texture for desert bushes
    TEXTURE_MAP {
        IMAGE_FILENAME = gaussian.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 1.0
        MATERIAL_ID = 28
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }

    # Texture for deciduous #1
    TEXTURE_MAP {
        IMAGE_FILENAME = gaussian.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 1.0
        MATERIAL_ID = 31
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }

    # Texture for deciduous #2
    TEXTURE_MAP {
        IMAGE_FILENAME = gaussian.pgm
        INSERT_POINT = 25000,75000,0
        GSD = 1.0
        MATERIAL_ID = 33
        MINIMUM_WAVELENGTH = 0.45
        MAXIMUM_WAVELENGTH = 0.52
    }
}

PLUME {
    BLOB_MODEL {
        VOXEL_FILENAME = ../../vxl_files/desertD/fbm_blob100.vxl
        #INSERT_POINT is used to override the voxel file x, y, z coordinates
        # INSERT_POINT = 1017.0, 2222.0, 0.0
        EXIT_TEMPERATURE = 350.0
    }
    GAS_LIST {
        GAS {
            NAME = SOMAN Gas

```

```
ABSORPTION_FILENAME = gd.gas  
RELEASE_MASS = 1.0  
MOLECULAR_WEIGHT = 182.2  
    }  
  }  
}
```

12 Appendix C: Creating Animations

This section describes the process of creating animations from DIRSIG images.

The first step is to create a series of voxel files over a given time range. Altering the blob.dat file time parameters does this. I.e., to make an animation from 25 seconds to 225 seconds after release at 10 second intervals set start time = 15 , end time = 225 and time_inc = 21. This will produce 21 blob???.vxl files. The question marks will be replaced by the time in seconds after release, i.e., blob25.vxl, blob35.vxl, blob45.vxl...blob225.vxl. Proceed to create fBm voxel files by running fBm.pro. Using the above scenario this will produce 21 fbm_blob???.vxl files. The starting coordinates found in blob.dat must be chosen from actual scene coordinates. To determine actual scene coordinates the user must run DIRSIG for a scene, enabling HIT_MAPS = TRUE under the TRUTH_IMAGES section. The user can then use ENVI to load the x, y, and z hitmaps as an RGB image. The user then selects from the ENVI menus: Function → Interactive Analysis → Cursor Location / Value to determine the x, y, z location of each pixel in scene units.

The second step is to run DIRSIG using the same scene with the gas model enabled. For this scenario the same scene would run in DIRSIG 21 times. The only difference between each run is the fbm_blob???.vxl file pointed to by the PLUME section of the configuration file. The user should set RESPONSE_FILENAME = IDEAL located in the BAND section of the configuration file. This will ensure that the output image is only one band. The user should also give each output image a different name. This can be set under the BAND section as IMAGE_FILENAME = *user_defined.img* and under the TRUTH_IMAGES section as IMAGE_FILENAME = *user_defined.img* in the configuration file.

The third step is to convert each image file, extension .img, to a tag image file format (extension .tiff) image. This was done using a utility at the DIRS lab called `make_displayable`. Its usage is: `make_displayable -t input_file output_file gain bias`. Setting the gain equal to a negative number autoscales the output into an 8-bit range. `Make_displayable` was used on the first image of each time sequence in autoscaling mode. This provided a gain and bias that was used on all subsequent images in that time sequence.

The fourth step is to take the scaled tiff images and use the freeware package Image Magick to create an animation. This is done with the following command in Image Magick:

```
>animate *.tiff
```

To create an animated GIF of the sequence use the following commands in Image Magick:

```
>convert -delay 20 *.tiff gas_animate.gif
```

C-shell scripts were used to do steps 2 and 3. It is highly suggested to the user that scripts be created to setup and run steps 2 and 3. It will save the user a great deal of time.

13 Appendix D: Adding Spectral Noise in ENVI

13.1 Adding Spectral Noise Using EigenVector-Based Transforms

All systems are subject to noise. Noise will have the effect of making cloud detection more difficult especially for lower gas cloud concentrations. Hence noise should be added to the final DIRSIG gas cloud scenes in order create more realistic scenarios for detection algorithm testing.

The idea behind eigenvector transforms is to redefine the feature space so the correlation between elements of a random vector (in this case spectral bands) is reduced. One of the most common of these transforms is known as principal component analysis. The multi-spectral information is translated and rotated to increase separability and maximize the variance. The new output feature is known as a principal component (PC). The number of PCs will be equal to the number of original bands, i . The first PC_1 will contain the largest percentage of total scene variance and each succeeding component PC_2, PC_3, \dots, PC_i will have decreasing variance. Since each PC is orthonormal to one another each subsequent band has no correlation with the next. Figure 13-1 illustrates.

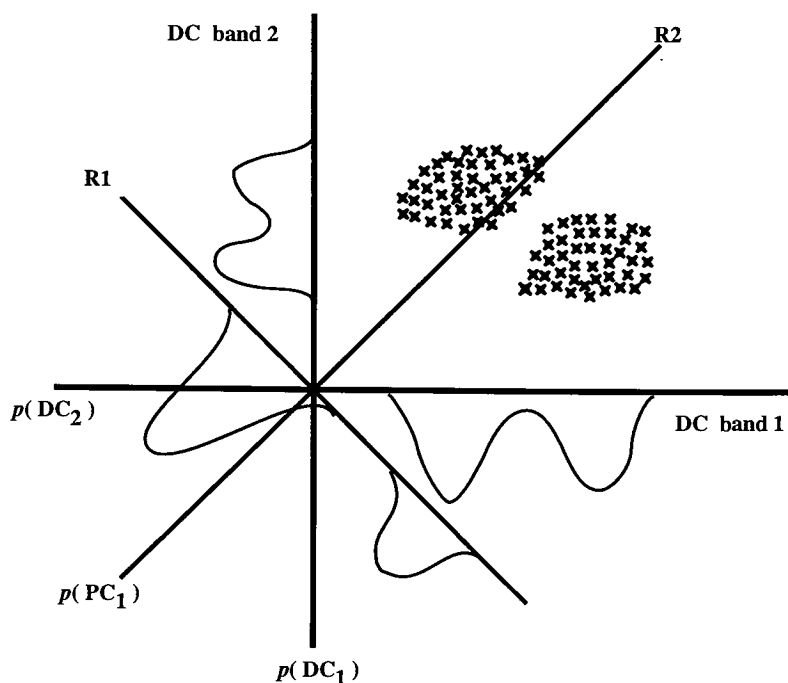


Figure 13-1 Increased separability and variance due to the projection onto a rotated principle component axis [Schott, 1997]

Principal component analysis can be used in the following manner to add spectrally correlated noise, obtained from dark current images, to multi-spectral image data:

1. Take the covariance matrix of the dark current image: **A**.
2. Take original multi-spectral image and calculate the covariance matrix: **B**.
3. Add the dark current covariance matrix and the multi-spectral covariance matrix together: $\mathbf{M} = \mathbf{A} + \mathbf{B}$.
4. The covariance matrices **A**, **B**, and **M** are used to generate 3 new corresponding matrices **A''**, **B''**, and **M''** where the rows, R_1, R_2, \dots, R_i are ordered eigenvectors corresponding to the ordered eigenvalues of each covariance matrix: **A**, **B**, and **M**. For convenience, the rows are arranged in order of decreasing magnitude of the corresponding eigenvalues. Thus **A''** is defined such that:

$$A''AA''^T = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \ddots & 0 \\ 0 & 0 & \lambda_i \end{bmatrix}$$

Equation 13-1

Where A''^T = the transpose of A'' , A = the covariance matrix, λ_i = eigenvalues of A and in addition represent the variance of the i th PC.

5. Use principal components analysis (forward transform) with matrix B'' to decorrelate the image into PCs:

$PC = B''^T X$, where

$$PC = \begin{bmatrix} PC_1 \\ \vdots \\ PC_i \end{bmatrix}$$

Equation 13-2

Where PC = principal component of the i th band and X = vector containing i digital counts by i spectral bands

6. Use principal components analysis (back transform) using matrix M'' to obtain original multi-spectral image plus spectrally correlated noise:

$$X = (M''^T)^{-1}PC$$

Equation 13-3

For a more rigorous treatment of principal component analysis see Schott (1997) and Castleman (1996).

The remote sensing software ENVI 3.0b, June 23rd, 1998 copyrighted and supported by Research Systems, Inc³. can be used to apply spectrally correlated noise.

³ <http://www.rsinc.com/envi>

13.2 Adding Spectrally Correlated Noise Using a Dark Current Image

If a corresponding dark current image is available then the following procedure can be used. The text presented here is taken from the ENVI online help and the ENVI User's Guide version 2.0, July 1995 edition.

A. The first step is to create a covariance matrix of the dark current image:

The Statistics option under the Basic Tools menu allows you to generate statistical reports for image files. The covariance option (which includes eigenvectors, the covariance matrix, and the correlation matrix) can be calculated for multi-band images only. The statistics are calculated in double precision.

1. Select Basic Tools > Statistics > Compute Statistics.

The Compute Statistics Input File dialog will appear. You will select input data files using the standard ENVI input file dialogs.

2. Select the dark current image by clicking on the file name in the list labeled "Select Input File".

4. Once the file has been selected, click "OK" to continue.

The Compute Statistics Parameters dialog will appear. You will use the Compute Statistics Parameters dialog to select the type of statistics to be calculated and the types of reports to be generated.

5. Select the "Covariance" check boxes.

Save the statistics to an ENVI format statistics file for later use with the ENVI function Principal Components . The default file extension for statistics files is .sta.

6. Click "OK" to start the statistics calculations.

A small window appears showing the percent of processing complete as a slider from 0 to 100%. The statistics file will contain the covariance and correlation matrices along with the eigenvectors for each band.

B: Repeat A with multi-spectral image file:

C: Add covariance of the dark noise to covariance of the multi-spectral image. Add eigenvectors of the dark noise to the eigenvectors of the multi-spectral image:

To view statistics previously saved by the "Compute Statistics" function (as .sta files) or other functions that create statistics files (e.g., "Principal Components").

1. Select Basic Tools > Statistics > View Statistics File.

2. When the standard ENVI file input dialog appears, select a valid statistics file.

3. Select File > Save Text to ASCII.

4. Outside of ENVI, i.e., in EXCEL, add the dark noise covariance matrix to the multi-spectral covariance matrix. Repeat for the eigenvectors. Save the new file.

D: Decorrelate multi-spectral image using principal components analysis:

PC Rotation from Existing Stats

If you have already calculated covariance and eigenvalue statistics for your data, you may use them as input into the PC Rotation. You may use any statistics file in the PC rotation that contains covariance and eigen statistics for the same number of bands as your input data.

1. Select Transforms > Principal Components > Forward PC Rotation > PC Rotation from Existing Stats.

2. When the standard ENVI file selection and subsetting dialogs appear, select and subset your input file using the standard ENVI file selection procedures.

Another file selection dialog will appear, which lists all of the existing statistics file (with a default file extension of .sta by default) in the current input data directory.

3. Select the statistics file with standard ENVI file selection procedures.

4. Select "File" or "Memory" output.

If "File" output is selected, enter a filename in the "Enter Output Filename" text box or use the "Choose" button to select a filename.

5. Select the desired type from the "Output Data Type" pulldown menu to control the data type of the output file.

6. Click "OK" to perform the forward PC rotation.

When ENVI has finished processing, it will load the PC bands into the Available Bands List where they may be displayed, and the PC EigenValues plot window will appear.

E: Recorrelate, decorrelated multi-spectral image using the covariance and eigenvectors from part C:

To transform principal component images back into their original data space:

1. Select Transforms > Principal Components > Inverse PC Rotation.

2. When the standard ENVI file selection and subsetting dialogs appear, select and subset your input file using the standard ENVI file selection procedures.

Another file selection dialog will appear, which lists all of the existing statistics file (with a default file extension of .sta by default) in the current input data directory.

3. Select the **part C** statistics file using standard ENVI file selection procedures.

Note: The statistics file must already exist prior to selecting the inverse PC rotation.

4. Select "File" or "Memory" output.

If "File" output is selected, enter a filename in the "Enter Output Filename" text box or use the "Choose" button to select a filename.

5. Select the desired type from the "Output Data Type" pulldown menu to control the data type of the output file.

6. Click "OK" to perform the inverse transform.

7. The output will be a multi-spectral image with spectrally correlated noise.

When ENVI has finished processing, it will load the bands into the Available Bands List where they may be displayed using standard gray-scale or RGB color composite methods.

13.3 Adding Spectrally Correlated Noise Without a Dark Current Image

This section discusses a way in ENVI to estimate the spectral noise statistics when a dark current image is not available. This method uses the minimum noise fraction transform (MNF). See Green (1988) and Boardman (1994) for additional information.

1. Select Transforms > MNF Rotation > Forward MNF > Estimate Noise Statistics From Data.

Use this option to estimate noise when you have no dark current image for your data, as is usually the case. ENVI assumes that each pixel contains both signal and noise, and that adjacent pixels will contain the same signal, but different noise. A "shift difference" is performed on the data by differencing adjacent pixels to the right and above each pixel and averaging the results to obtain the "noise" value to assign to the pixel being processed. The best noise estimate is gathered using the shift-difference statistics from a homogeneous area rather than from the whole image. ENVI will allow you to select the subset for statistics extraction.

2. When the standard ENVI file selection and subsetting dialogs appear, select and subset your input file using the standard ENVI file selection procedures.

3. When the Forward MNF Transform Parameters dialog appears, enter a filename for the noise statistics in the text box labeled "Enter Output Noise Stats Filename [.sta]."

To select a homogeneous area for calculating the noise statistics, use the "Spatial Subset" button to use ENVI's standard spatial subset procedures to either manually enter a subset or to graphically indicate the area for statistics extraction.

4. Enter an output file for the MNF statistics in the text box labeled "Enter Output MNF Stats Filename [.sta]."

Caution: Be sure that the MNF and noise statistics files have different names.

5. Select "File" or "Memory" output.

If "File" output is selected, enter a filename in the "Enter Output Filename" text box or use the "Choose" button to select a filename.

6. Click "OK" to start the processing.

7. Use the covariance and eigenvectors from the noise statistics file as the dark current image covariance and dark current image eigenvectors.

8. All other steps are the same as adding spectrally correlated noise using a dark current image as outlined in section 13.2.

When ENVI has finished processing, it will load the MNF bands into the Available Bands List.

The MNF Eigenvalue plot window will appear. Bands with large eigenvalues (greater than one) contain data, and bands with eigenvalues near one contain noise

14 Appendix E: Design Experiment for Truth Data Via a Benign Gas

14.1 Design Outline

I. Objective

To outline the steps needed to collect concentration, dispersion, and rise data of a benign gas in order to validate the dispersion model presented in this research.

II. Equipment

1. A benign gas with similar physical properties to the toxic agent to be studied. I.e., similar molecular weight and infrared absorption characteristics. It is also preferable to have a gas that is visible to the human eye because then visual notes can be taken along with a video of the dispersal.
2. Spectrometers to measure radiance profiles of the gas release at various points down wind over the wavelengths of interest.
3. A controlled environment, such as a wind tunnel with a heating element. This allows for varying wind speeds, temperatures, and turbulent conditions.
4. Control over parameters which effect the atmosphere would also be beneficial. These parameters include humidity, aerosol content, etc...
5. The inclusion of objects such as models of buildings, vegetation, vehicles, etc... would help in the study of object interaction with the cloud.
6. Additional experimentation could include releases outside using large outdoor model environments such as in parking lots, desert scenes, or base camps. These

experiments would be more difficult to control due to parameters such as changing wind directions but offer more realistic results.

III. Experiment

1. In a controlled environment such as a wind tunnel set up spectrometers and video cameras to record data.
2. Vary wind speed and insolation to cover all Pasquill classes. Record data at varying distances and times after release. Measure concentration, absorption, and rise of gas cloud. Observe cloud dispersion and measure widths at varying distances.

IV. Compare Results

1. Use collected truth data to verify DIRSIG modeled data.
2. Use a metric to quantify differences between truth and modeled data.
3. Comment on usefulness and accuracy of modeled data.

V. Make recommendations on improvements to the model

1. Comment on possible improvements or scaling factors to make model more accurate.
2. List any problems encountered in the validation process.
3. Document all findings in a final report

15 References

- Babrauskas, Vytenis, *Toxicity for the Primary Gases Found in Fires* [online], 1997, Fire Science and Technology Inc. Available from: <http://www.blarg.net/~fsti/toxicity.html> [Accessed 10/12/99]
- Berk A., Benstein L. S., and Robertson, *MODTRAN: A Moderate Resolution Model for LOWTRAN 7*, GL-TR-89-0122, Spectral Sciences, D.C., 1989.
- Beychock, M., *Fundamentals of Stack Gas Dispersion*, Irvine, CA, 3rd ed., 1994.
- Beychock, M., *Error Propagation in Air Dispersion Modeling* [online]. Available from: <http://www.air-dispersion.com/default.htm> [Accessed 09/16/99]
- Blackadar, Alfred, *Turbulence and Diffusion in the Atmosphere*, Springer, NY, 1997.
- Boardman J. W., and Kruse F. A., Automated Spectral Analysis: a geological example using AVIRIS data, north Grapevine Mountains, Nevada, *Proceedings, ERIM Tenth Thematic Conference on Geologic Remote Sensing*, Environmental Research Institute of Michigan, Ann Arbor, MI, p I-407 – I-418, 1994.
- Briggs G. A., *Plume Rise*, AEC Critical Review Series, 1969.
- Brown, Scott D., *DIRSIG Users' Manual Release 3.1.1*, Rochester Institute of Technology, Rochester, 1999.
- Castleman, Kenneth R., *Digital Image Processing*, Prentice-Hall, New Jersey, 1996.
- Crownover, Richard M., *Introduction to Fractal and Chaos*, Jones And Bartlett, Boston, 1995.
- DCS Corporation, *AIRSIM thermal signature and prediction analysis tool definition and analysis of object inputs*, DCS Technical Note 9090-002-004, DCS Corporation, December, 1990.
- Dereniak E. L., Crowe D. G., *Optical Radiation Detectors*, John Wiley & Sons, 1984.

Falconer, K., *Fractal Geometry, Mathematical Foundations, and Applications*, Wiley, Chichester, 1990.

Fujimoto, Akira, ARTS: Accelerated Ray-Tracing System, *IEEE*, p 16-26, April, 1986.

Fusner, Charles, *Basic Raytracing F.A.Q.: A Slightly More Indepth Explanation* [ONLINE], 1999. Available from: <http://www.enter.net/~cfusner/tutorial/ch02.html> [Accessed 10/13/1999]

Gifford, Hangen, D.A. (editor), *Lectures on Air Pollution and Environmental Impact Analyses*, American Meteorological Society, Sept 1975.

Goody R. M., and Yung Y. L., *Atmospheric Radiation*, Oxford University Press, 1989.

Green A. A., Berman M., Switzer P., Craig M. D., A Transformation for Ordering Multispectral data in Terms of Image Quality with Implications for Noise Removal, *IEEE Transactions on Geoscience and Remote Sensing*, **26**(1), p 65-74, 1988.

Grum F., Becheru R. J., *Optical Radiation Measurements: Vol.1, Radiometry*, Academic, NY, 1979.

Heckbert, Paul (editor), *Graphics Gems IV*, Academic Press, p 366-369, 1994.

Hoffland L., Piffath R., Bouck J., Spectral Signatures of Chemical Agents and Simulants, *Optical Engineering*, **24**(6), p 982-984, Nov/Dec, 1985.

ICA division, (1997), *Nerve Agents* [online]. Available from: <http://www.opcw.nl/chemhaz/nerv.htm> [Accessed 08/30/99]

Kerekes, J., Landgrebe D., A Noise Taxonomy for Remote Sensing Systems, *Proceedings of IGARSS '87 Symposium*, Ann Arbor, p 903-908, 18-21 May 1987.

Kuo, Didi, S., *Synthetic Image Generation of Factory Stack and Water Cooling Tower Plumes*, Ph.D. Dissertation, Rochester Institute of Technology, Dept. of Imaging Science, NY, Aug, 1997.

Lillesand T., Kiefer R., *Remote Sensing and Image Interpretation*, John Wiley and Sons, NY, 1994.

- Lovejoy S., Mandelbrot B., Fractal Properties of Rain and a Fractal Model, *Tellus*, **37A(3)**, p 209-332, 1985.
- Mandelbrot B., On the Geometry of Homogeneous Turbulence, with Stress on the Fractal Dimension of the Isosurfaces of Scalars, *Journal of Fluid Mechanics*, **72(2)**, p 401-416, 1975.
- McMullen R. W., The Change of Concentration Standard Deviations with Distance, *JAPCA*, Oct, 1975.
- Newman W., Sproull R., *Principles of Interactive Computer Graphics*, McGraw-Hill, 1979.
- Peitgen H., Saupe Dietmer, *The Science of Fractal Images*, Springer-Verlag, 1986.
- Prandtl L., Über Die Ausgebildete Turbulenz, *Zeitschr, Fur Angew, Math u Mech*, **5**, p 136, 1925.
- Sakas G., Modeling and Animating Turbulent Gaseous Phenomena Using Spectral Synthesis, *The Visual Computer*, **9**, p 200-212, 1993.
- Schmidt W., *Der Massenaustausch in freier Luft und verwandte Erscheinungen*, Henri Grand, Hamburg, 1925.
- Schott J. R., *Remote Sensing: The Image Chain Approach*, Oxford, NY, 1997.
- Slade D.H. (editor), *Meteorology and Atomic Energy 1968*, Air Resources Laboratories, U.S. Department of Commerce, July, 1968.
- Smith H.J.P., Dube D.J., Gardner M.E., Clough S.A., Kneizys F.X., and Rothman L.S., *FASCODE – Fast Atmospheric Signature Code (Spectral Transmittance and Radiance)*, Air Force Geophysics Laboratory Report AFGL-TR-78-0081, Hanscom AFB, MA, 1978.
- Sreenivasan K., Fractals and Multifractals in Fluid Turbulence, *Annual Revue Fluid Mech*, **23**, p 539-600, 1991.
- Stam Jose, *A Multi-Scale Model for Computer Graphics*, MS Thesis, University of Toronto, Dept of Computer Science, Toronto, 1991.
- Stam Jose, *Multi-Scale Stochastic Modeling of Complex Natural Phenomena*, Ph.D. Dissertation, University of Toronto, Dept of Computer Science, Toronto, 1995.

Taylor G. I., Statistical Theory of Turbulence, Part I, *Proc Roy Soc London*, Ser. A., **151**, p 421-444, 1935.

Tompson A. F. B., Ababou R., Gelhar L., Implementation of the Three-Dimensional Turning Bands Random Field Generator, *Water Resources Research*, **25(10)**, p 2227-2243, Oct, 1989.

Yaglom A. M., *Correlation Theory of Stationary and Related Random Functions I: Basic Results*, Springer-Verlag, 1986.

Yin Z. M., New Methods for Simulation of Fractional Brownian Motion, *Journal of Computational Physics*, **127**, p 66-72, 1996.